Hujun Yin Ke Tang Yang Gao
Frank Klawonn Minho Lee Bin Li
Thomas Weise Xin Yao (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2013

LNCS 8206

14th International Conference, IDEAL 2013
Hefei, China, October 2013
Proceedings

Springer

Lecture Notes in Computer Science 8206

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

Hujun Yin Ke Tang Yang Gao
Frank Klawonn Minho Lee Bin Li
Thomas Weise Xin Yao (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2013

14th International Conference, IDEAL 2013
Hefei, China, October 20-23, 2013
Proceedings

Springer

Volume Editors

Hujun Yin
The University of Manchester, UK; E-mail: hujun.yin@manchester.ac.uk

Ke Tang
Bin Li
Thomas Weise
University of Science and Technology of China, Hefei, China
E-mail: {ketang, tweise, binli}@ustc.edu.cn

Yang Gao
Nanjing University, China; E-mail: gaoy@nju.edu.cn

Frank Klawonn
Ostfalia University of Applied Sciences, Wolfenbüttel, Germany
E-mail: f.klawonn@ostfalia.de

Minho Lee
Kyungpook National University, Daegu, Korea; E-mail: mholee@knu.ac.kr

Xin Yao
University of Birmingham, UK; E-mail: x.yao@cs.bham.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41277-6 e-ISBN 978-3-642-41278-3
DOI 10.1007/978-3-642-41278-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013948928

CR Subject Classification (1998): I.2.6-7, H.2.8, I.5, I.2.10-11, J.3, F.2.2, H.3, F.1, H.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

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

Printed on acid-free paper

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

Preface

In this digital era and subsequent "data rich but information poor" quandary, the IDEAL conference serves its purposes perfectly – making sense of huge volumes of data, evaluating the complexity of real-world problems, and turning data into information and knowledge. The IDEAL conference attracts international experts, researchers, leading academics, practitioners, and industrialists from communities of machine learning, computational intelligence, data mining, knowledge management, biology, neuroscience, bio-inspired systems and agents, and distributed systems. It has enjoyed a vibrant and successful history in the last 15 years, having been held in over 11 locations in 7 different countries. It continues to evolve to embrace emerging topics and exciting trends. This year IDEAL set foot in mainland China, the fastest growing economy in the world. The conference received about 130 submissions, which were rigorously peer-reviewed by Program Committee members. Only the papers judged to be of highest quality were accepted and included in these proceedings.

This volume contains 76 papers accepted and presented at the 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2013), held on 20–23 October 2013 in Hefei, China. These papers provided a valuable collection of recent research outcomes in data engineering and automated learning, from methodologies, frameworks, and techniques to applications. In addition to various topics such as evolutionary algorithms; neural networks; probabilistic modelling; swarm intelligent; multi-objective optimization and practical applications in regression, classification, clustering, biological data processing, text processing, and video analysis, IDEAL 2013 also featured a number of special sessions on emerging topics such as adaptation and learning multi-agent systems, big data, swarm intelligence and data mining, and combining learning and optimization in intelligent data engineering.

We would like to thank all the people who devoted so much time and effort to the successful running of the conference, in particular the members of the Program Committee and reviewers, as well as the authors who contributed to the conference. We are also very grateful for the hard work of the local organizing team at the University of Science and Technology of China (USTC), especially Prof. Bin Li, in local arrangements, as well as the help provided by the University

of Manchester in checking through all the camera-ready files. Continued support and collaboration from Springer, in particular from the LNCS editor, Alfred Hoffman and Anna Kramer, are also greatly appreciated.

August 2013

Hujun Yin
Ke Tang
Yang Gao
Frank Klawonn
Minho Lee
Bin Li
Thomas Weise
Xin Yao

Organization

General Chair

Xin Yao	University of Birmingham, UK

Program Chair

Hujun Yin	University of Manchester, UK

Program Co-chairs

Ke Tang	University of Science and Technology of China (USTC), China
Yang Gao	Nanjing University, China
Frank Klawonn	Ostfalia University of Applied Sciences, Germany
Minho Lee	Kyungpook National University, South Korea

International Advisory Committee

Lei Xu (Chair)	Chinese University of Hong Kong, Hong Kong
Yaser Abu-Mostafa	CALTECH, USA
Shun-ichi Amari	RIKEN, Japan
Michael Dempster	University of Cambridge, UK
José R. Dorronsoro	Autonomous University of Madrid, Spain
Nick Jennings	University of Southampton, UK
Soo-Young Lee	KAIST, South Korea
Erkki Oja	Helsinki University of Technology, Finland
Latit M. Patnaik	Indian Institute of Science, India
Burkhard Rost	Columbia University, USA
Xin Yao	University of Birmingham, UK

Steering Committee

Hujun Yin (Chair)	University of Manchester, UK
Laiwan Chan (Chair)	Chinese University of Hong Kong, Hong Kong
Guilherme Barreto	Federal University of Ceará, Brazil
Yiu-ming Cheung	Hong Kong Baptist University, Hong Kong

Emilio Corchado	University of Burgos, Spain
Jose A. Costa	Federal University of Rio Grande do Norte, Brazil
Colin Fyfe	University of the West of Scotland, UK
Marc van Hulle	K.U. Leuven, Belgium
Samuel Kaski	Helsinki University of Technology, Finland
John Keane	University of Manchester, UK
Jimmy Lee	Chinese University of Hong Kong, Hong Kong
Malik Magdon-Ismail	Rensselaer Polytechnic Inst., USA
Vic Rayward-Smith	University of East Anglia, UK
Peter Tino	University of Birmingham, UK
Zheng Rong Yang	University of Exeter, UK
Ning Zhong Maebashi	Inst. of Technology, Japan

Publicity Co-chairs

Emilio Corchado	University of Salamanca, Spain
Jose A. Costa	Federal University of Rio Grande do Norte, Brazil
Thomas Weise	USTC, China

International Liaison

Jinlong Li	USTC, China
David Camacho	Universidad Autónoma de Madrid, Spain
Guilherme Barreto	Federal University of Ceará, Brazil
Brijesh Verma	Central Queensland University, Australia

Local Organizing Committee

Bin Li (Chair)	University of Science and Technology of China
Kaiming Chen	University of Science and Technology of China
Jinlong Li	University of Science and Technology of China
Thomas Weise	University of Science and Technology of China
Rui Xu	University of Science and Technology of China
Ke Tang	University of Science and Technology of China

Program Committee

Ajith Abraham	Romis Attux
Davide Anguita	Javier Bajo Pérez
Bruno Apolloni	Bruno Baruque
Francisco Assis	Carmelo J.A. Bastos Filho

Vicent Botti
Antonio Braga
Fernando Buarque
Andrés Bustillo
José Luis Calvo Rolle
David Camacho
Heloisa Camargo
Anne Canuto
Jaime Cardoso
Andre Carvalho
Matthew Casey
Darryl Charles
Richard Chbeir
Chunlin Chen
Ling Chen
Songcan Chen
Tianshi Chen
Sung-Bae Cho
Seungjin Choi
Stelvio Cimato
David Clifton
André Coelho
Leandro Coelho
Francesco Corona
Paulo Cortez
Marcelo A. Costa
Raúl Cruz-Barbosa
Ernesto Cuadros-Vargas
Alfredo Cuzzocrea
Ernesto Damiani
Nicoletta Dessì
Fernando Díaz
Weishan Dong
Jose Dorronsoro
Adrião Duarte
Jochen Einbeck
Pablo Estevez
Igor Farkas
Jan Feyereisl
Felipe M.G. França
Richard Freeman
Kunihiko Fukushima
Colin Fyfe
Marcus Gallagher
Laura Garcia
Ana Belén Gil
Fernando Gomide
Petro Gopych
Marcin Gorawski
Juan Manuel Górriz
Lars Graening
Manuel Grana
Maciej Grzenda
Arkadiusz Grzybowski
Jerzy Grzymala-Busse
Alberto Guillen
Barbara Hammer
Ioannis Hatzilygeroudis
Francisco Herrera
Álvaro Herrero
J. Michael Herrmann
James Hogan
Jaakko Hollmén
Vasant Honavar
Wei-Chiang Samuelson Hong
David Hoyle
Estevam Rafael Hruschka Junior
Naohiro Ishii
Konrad Jackowski
Gil-Jin Jang
He Jiang
Ata Kaban
Juha Karhunen
Rheeman Kil
Dae-Shik Kim
Dong-Joo Kim
Kee-Eung Kim
Kyunghwan Kim
Kyung-Joong Kim
Mario Koeppen
Andreas König
Bartosz Krawczyk
Pavel Kromer
Rudolf Kruse
Sungoh Kwon
Pei Ling Lai
Jonghwan Lee
Seong-Whan Lee
Soo-Young Lee
Ming Li

Jongjin Lim
Paulo Lisboa
Jing Liu
Wenjian Luo
Jiancheng Lv
José Everardo B. Maia
Roque Marin Morales
Urszula Markowska Kaczmar
Jan Martinovic
Giancarlo Mauri
José Ramón Méndez
Yoan Miche
Byoung-Kyong Min
José Manuel Molina
Petr Musilek
Tim Nattkemper
Antonio Neme
He Ni
Yusuke Nojima
Erkki Oja
Chung-Ming Ou
Seiichi Ozawa
Vasile Palade
Hyeyoung Park
Hyung-Min Park
Juan Pavón
Carlos Pedreira
Jan Platos
Jose Principe
Héctor Quintian
Yacine Rebahi
Raquel Redondo
Agnaldo Reis
Bernardete Ribeiro
Ignacio Rojas
Jesús Ángel Román
Fabrice Rossi
Sherif Sakr
Luciano Sánchez Ramos
Javier Sedano
Jose Seixas
Yinghuan Shi
Catarina Silva
Hernando Silva
Ivan Silva
Dragan Simic
Olli Simula
Michael Small
Piotr Sobolewski
Alessandro Sperduti
Ilhong Suh
Ying Tan
Ricardo Tanscheit
Dante Tapia
Chuan-Kang Ting
Peter Tino
Renato Tinós
Alicia Troncoso
Eiji Uchino
Marc van Hulle
Belen Vaquerizo
Marley Vellasco
Alfredo Vellido Alcacena
Vicente Vera
Michel Verleysen
Fernando Von Zuben
Jian-Wu Wan
Jiahai Wang
Lipo Wang
Wenjia Wang
Yong Wang
Katarzyna Wegrzyn-Wolska
Dongqing Wei
Thomas Weise
Michal Wozniak
Zhenyu Yang
Takashi Yoneyama
Yang Yu
Du Zhang
Min-Ling Zhang
Aimin Zhou
Huiyu Zhou
Zexuan Zhu
Rodolfo Zunino

Special Session on Adaptive and Learning Multi-agent Systems

Organizers

Hongbin Dong	Harbin Engineering University, China
Jun He	Aberystwyth University, UK
Xinjun Mao	National University of Defense Technology ,China
Xiangrong Tong	Yantai University, China

Special Session on Big Data

Organizers

Hui Xiong	Rutgers University, USA
Wenjun Zhou	University of Tennessee, USA

Special Session on Soft-Computing Algorithms in Renewable Energy Problems

Organizers

Sancho Salcedo Sanz	Universidad de Alcalá, Madrid, Spain
Jose Antonio Portilla-Figueras	Universidad de Alcalá, Madrid, Spain

Special Session on Swarm Intelligence and Data Mining

Organizers

Jing Liang	Zhengzhou University, China
Chuan-Kang Ting	National Chung Cheng University, Taiwan
Ying-ping Chen	National Chiao Tung University, Taiwan

Special Session on Text Data Learning

Organizers

Baoli Li	Henan University of Technology, China
Carl Vogel	Trinity College Dublin, Ireland

Special Session on Coevolution

Organizers

Siang Yew Chong	University of Nottingham, Malaysia Campus
Zhenyu Yang	National University of Defense Technology, China
Xiaodong Li	Royal Melbourne Institute of Technology, Australia

Special Session on Combining Learning and Optimization for Intelligent Data Engineering

Organizers

Ata Kaban	The University of Birmingham, UK
Jakramate Bootkrajang	The University of Birmingham, UK

Table of Contents

A Pruning Algorithm for Extreme Learning Machine

Li Ying[1] and Li Fan-jun[2]

[1] School of Science, Qi Lu University of Technology, Jinan, Shandong, China 250353
[2] School of Mathematical Science, University of Jinan, Shandong, China 250022
lzhbb07@sina.com, ss_lifj@ujn.edu.cn

Abstract. It is difficult for Extreme Learning Machine (ELM) to estimate the number of hidden nodes used to match with the learning data. In this paper, a novel pruning algorithm based on sensitivity analysis is proposed for ELM. The measure to estimate the necessary number of hidden layer nodes is presented according to the defined sensitivity. When the measure is below the given threshold, the nodes with smaller sensitivities are removed from the existent network all together. Experimental results show that the proposed method can produce more compact neural network than some other existing similar algorithms.

Keywords: Single-hidden-layer feedforward neural networks, Extreme Learning Machine, Sensitivity analysis, Pruning algorithm.

1 Introduction

For single-hidden layer feedforward networks (SLFNs), Huang et al. proposed an original algorithm called extreme learning machine (ELM), which has been shown good generalization performance at extremely high learning speed in most cases [1-2]. However, the number of hidden nodes to be used for ELM remains a trial and error process. How to automatically produce a compact network for given data is one of the open problems in ELM research [5, 12, 13].

Two different approaches are often used to solve this problem. The first is constructive algorithms [3-6], which make it easy for the user to determine the network structure. However, it has been proved that the learning capacity of feedforward neural network is proportional to its complexity [6-8]. As the model complexity increases, with all other factors hold constant, the generalization performance of the neural networks improves up to a certain limit, after that it will deteriorate [12, 13].

Another alternative is pruning algorithms, which firstly train a larger neural network than necessary, then remove the unnecessary nodes during or after learning [9-13]. Rong et al. presented a pruned ELM (P-ELM) as an automated method for ELM design [12]. The P-ELM algorithm calculates the relevance between hidden nodes and the class labels using statistical criterions, and eliminates the hidden nodes with lower relevance to produce compact neural networks, which generate competitive prediction accuracy on unseen data. The P-ELM algorithm mainly focuses on pattern classification problems. Another pruning algorithm called optimally-pruned ELM (OP-ELM) was proposed by Miche et al. for both regression and classification applications [13].

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 1–7, 2013.

Experimental results illustrate that OP-ELM is only a compromise between the speed of the ELM and the accuracy of much slower and complicated methods [13].

In this paper, the pruned extreme learning machine based on sensitivity analysis (SAP-ELM) is presented. Section 2 briefly reviews the SLFN model and the original ELM. The details of the proposed algorithm are described in Section 3. Section 4 presents the experiment results. Finally, Section 5 summarizes the main conclusions.

2 Brief of SLFN and ELM

For given N arbitrary distinct samples $\{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in R^n, \boldsymbol{t}_i \in R^m, 1 \le i \le N\}$, The SLFN with L randomly generated hidden nodes and activation functions $g(x)$ is represented by

$$f_L(\boldsymbol{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i g(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}_j + b_i) = \boldsymbol{o}_j, 1 \le j \le N \tag{1}$$

Where $w_i=(w_{i1}, w_{i2},\ldots, w_{in})^{\mathrm{T}}$ is the weight vector linking the ith hidden node to the input layer. b_i is the threshold of the ith hidden node. $\boldsymbol{\beta}_i=(\beta_{i1}, \beta_{i2},\ldots, \beta_{im})^{\mathrm{T}}$ is the weight vector connecting the ith hidden node with the output layer. Activation functions $g(x)$ include the sigmoid functions as well as the radial basis etc.. According to [2], the parameters of SLFN can be adjusted such that

$$f_L(\boldsymbol{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i g(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, 1 \le j \le N \tag{2}$$

Customarily, the above N equations can be rewritten as the following matrix equation

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T} \tag{3}$$

Where

$$\boldsymbol{H} = \begin{bmatrix} g_1(\boldsymbol{x}_1) & \cdots & g_L(\boldsymbol{x}_1) \\ \vdots & \cdots & \vdots \\ g_1(\boldsymbol{x}_N) & \cdots & g_L(\boldsymbol{x}_N) \end{bmatrix}_{N\times L} \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\beta}_L^{\mathrm{T}} \end{bmatrix}_{L\times m} \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{t}_N^{\mathrm{T}} \end{bmatrix}_{N\times m} \tag{4}$$

Where $g_i(\boldsymbol{x}_j)=g(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}_j+b_i)$, H is the hidden layer output matrix of the SLFN [2]. The ith column of H is the outputs of the ith hidden node with respect to the inputs, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_N$. The jth row of H is the outputs of all the hidden neurons with respect to the input $\boldsymbol{x}_j$.

Algorithm ELM [2]: Given a training set $\{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in R^n, \boldsymbol{t}_i \in R^m, 1 \le i \le N\}$, activation functions $g(x)$, and hidden node number L,

Step 1: Randomly assign input weights $\boldsymbol{w}=[\boldsymbol{w}^{\mathrm{T}}_1, \boldsymbol{w}^{\mathrm{T}}_2,\ldots, \boldsymbol{w}^{\mathrm{T}}_L]^{\mathrm{T}}$ and bias $\boldsymbol{b}=[b_1, b_2, \ldots, b_L]^{\mathrm{T}}$;

Step 2: Calculate the hidden layer output matrix $\boldsymbol{H}$;

Step 3: Calculate the output weight matrix $\boldsymbol{\beta}=\boldsymbol{H}^{+}\boldsymbol{T}$, where $\boldsymbol{H}^{+}$ is the Moore-Penrose generalized inverse matrix of $\boldsymbol{H}$.

3 SAP-ELM Algorithm

3.1 Definitions of Sensitivity and Model Scale

Given $h_{ij}=g(w_i x_j+b_i)$, Eq.(2) can be rewritten as

$$t_j = h_{1j}\beta_1 + h_{2j}\beta_2 + \cdots h_{Lj}\beta_L \quad 1 \le j \le N \tag{5}$$

Suppose the hidden node with β_1 is deleted, the network output can be presented as

$$t'_j = h_{2j}\beta_2 + h_{3j}\beta_3 + \cdots h_{Lj}\beta_L \quad 1 \le j \le N \tag{6}$$

We have the residual error with respect to all the inputs as following

$$\left| t_j - t'_j \right| = \left| h_{1j} \right| \left| \beta_1 \right|, 1 \le j \le N \tag{7}$$

Eq.(7) illustrates that the residual error caused by deleting one hidden node is proportional to the node's output and the length of the weight vector linking to output layer. So, the residual error's sensitivity to hidden nodes can be defined as

$$S_i = \frac{1}{N}\sum_{j=1}^{N} \left| h_{ij} \right| \left| \beta_i \right| \qquad 1 \le i \le L \tag{8}$$

Hidden nodes can be ranked according to the importance

$$S'_1 \ge S'_2 \ge \cdots \ge S'_L \tag{9}$$

The measure to estimate the necessary number of hidden layer neurons is defined as

$$M_k = \frac{\sum_{i=1}^{k} S'_i}{\sum_{i=1}^{L} S'_i} \quad 1 \le k \le L \tag{10}$$

Eq.(10) illustrates that the bigger M_k means the more hidden nodes used. The necessary number of hidden layer neurons can be defined on the basis of M_k.

$$M = \min\left\{ k \mid M_k \ge \gamma, \quad 1 \le k \le L \right\} \tag{11}$$

Where γ ($0<\gamma<1$) is a given model scale threshold, M is the necessary number of hidden layer neurons.

3.2 Removing Nodes and Updating the Weights

Suppose k hidden nodes with weights β_i (i=1,2…,k) are removed together,

$$\left|t_j - t_j^k\right| = \left|h_{1j}\beta_1 + h_{2j}\beta_2 + \cdots + h_{kj}\beta_k\right|, 1 \le j \le N \tag{12}$$

Where t_j^k is the output of the network with respect to the input $\boldsymbol{x}_j$ after deleting k hidden nodes. h_{ij} is the output of the ith hidden node. Then, we have

$$\left|t_j - t_j^k\right| \le \sum_{i=1}^{k}\left|h_{ij}\right|\left|\beta_i\right|, 1 \le j \le N \tag{13}$$

Eq.(13) shows that the residual error caused by deleting k hidden nodes all together is not more than the sum of the residual errors caused by deleting the k hidden nodes one by one. So, SAP-ELM removes the redundant nodes all together to avoid a great amount of pruning and retraining computation.

We give the method to update the weights of the persisting nodes to remember the information of training sample stored in the deleted weights. According to Eq.(9) and Eq.(11), the subset of crucial hidden nodes is R= $\{j_1, j_2,\ldots, j_M\}$, another subset of deleting nodes is D= $\{d_1, d_2,\ldots, d_{L-M}\}$. The input weights of the crucial nodes are updated by the following formula:

$$w_{j_l,i}^{new} = w_{j_l,i}^{old} + \frac{1}{L-M}\sum_{k=1}^{L-M} w_{d_k,i}, \quad 1 \le i \le n, 1 \le l \le M \tag{14}$$

The output weights of the crucial nodes are updated by Moore-Penrose generalized inverse matrix of $\boldsymbol{H}'$:

$$\boldsymbol{\beta}' = (\boldsymbol{H}')^{+}\boldsymbol{T} \tag{15}$$

Where $\boldsymbol{H}'$ is the hidden layer output matrix of the pruned SLFN.

3.3 Steps of SAP-ELM

Given a training set $\{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in R^n, \boldsymbol{t}_i \in R^m, 1 \le i \le N\}$, activation functions, an initial larger number of hidden nodes than necessary, $L \le N$, and a threshold γ ($0 < \gamma < 1$),

Step 1: Construct a SLFN with L hidden nodes according to the original ELM,

Step 2: Calculate the residual error's sensitivity according to Eq.(8) and rank the hidden nodes similar to Eq.(9);

Step 3: Calculate the necessary number of hidden layer neurons on the basis of Eq.(10) and Eq.(11);

Step 4: Select the subset of deleting nodes according to Eq.(9) and Eq.(11);

Step 5: Update the weights using Eq.(14) and Eq.(15).

4 Performance Evaluation

4.1 Real-World Regression Problems

The approximation performance of SAP-ELM and original ELM are compared in this section. Table 1 shows the characteristics of the datasets [14]. The initial numbers of hidden nodes and the thresholds of SAP-ELM are list in Table 2. All the input and output values are normalized to [0, 1]. Fifty trials have been conducted under the same conditions for all cases and the average results are list in Table 3. As observed from Table 3, the SAP-ELM algorithm produces more stingy networks compared with the original ELM, while outperforming the ELM algorithm on the testing sets.

Table 1. Data used for regression problems

Data sets	Training	Testing	Attributes
Abalone	2000	2177	7
Triazines	100	86	60
Census(house8L)	10000	12784	8
Servo	80	87	4

Table 2. The values of parameter in regression problems

Data sets	Abalone	Triazines	Census(house8L)	Servo
No. of initial hidden nodes	500	100	500	80
Threshold	0.2	0.3	0.6	0.7

Table 3. Performance comparison in regression problems

Data sets	SAP-ELM			ELM		
	No. of final hidden nodes	Training error	Testing error	No. of final hidden nodes	Training error	Testing error
Abalone	18.28	0.0754	0.0784	25	0.0750	0.0803
Triazines	3.92	0.1945	0.1961	10	0.1785	0.2011
Census(house8L)	108.34	0.0632	0.0706	160	0.0612	0.0716
Servo	21.92	0.0796	0.1151	30	0.0623	0.1210

4.2 Real-World Classification Problems

Compared with the original ELM and the P-ELM, the classification performance of the proposed SAP-ELM have been tested on 4 real-world classification problems coming from UCI ML repository [14], which are summarized in Table 4. For all cases, the input values are normalized to [-1, 1]. Table 5 shows the initial numbers of hidden nodes and the thresholds. The χ^2 statistical method is used for pruning the insignificant hidden nodes in P-ELM with sigmoid function. The numbers of hidden nodes are gradually increased by an interval of 5 and the nearly optimal numbers of nodes for ELM are then

selected as done in [2]. The results shown in Table 6 are averages of 50 trials of simulations for each case. The winner in every comparing item has been shown in boldface in Table 6. According to Table 6, the proposed SAP-ELM produces the most compact networks among the three algorithms for all the cases, while keeping better generalization performance in most cases and obtains faster learning speed in very large complex applications than the original ELM and the P-ELM. The standard deviations of testing rate shown in Table 6 illustrates that both SAP-ELM and ELM obtain more stable generalization performance than P-ELM.

Table 4. Data used for classification problems

Data set	Training	Testing	Attribute	Class
S-image	4400	2035	36	6
L-recognition	10,000	10,000	16	26
Glass	110	104	9	7
Vehicle	420	426	18	4

Table 5. The values of parameter in classification problems

Data sets	S-image	L-recognition	Glass	Vehicle
No. of initial hidden nodes	500	500	100	200
Threshold	0.7	0.7	0.4	0.4

Table 6. Performance comparison in classification problems

Data sets	Approaches	No. of final hidden nodes	Training time (S)	Testing rate (%)	
				Ave	Dev
S-image	SAP-ELM	**264.66**	**25.693**	89.920	0.290
	ELM	500	85.941	89.770	**0.250**
	P-ELM[1]	397	49.533	**89.955**	0.563
L-recognition	SAP-ELM	**253.1**	**50.654**	**90.570**	**0.160**
	ELM	500	184.58	90.080	**0.160**
	P-ELM	483	214.03	89.669	0.212
Glass	SAP-ELM	**16.54**	0.1872	**68.850**	5.03
	ELM	30	**0.0624**	64.370	3.35
	P-ELM	18	0.5681	65.136	**3.32**
Vehicle	SAP-ELM	**35.5**	1.2012	**80.080**	**1.31**
	ELM	200	**0.6084**	79.540	1.70
	P-ELM	67	2.6883	79.254	2.25

[1] The results of P-ELM are same as the original papers for all cases.

5 Conclusions

This paper proposed a novel pruning algorithm for ELM based on sensitivity analysis. Whose goal is to automatically determine the number of the hidden nodes matching with learning data. The proposed approach weights the importance of hidden nodes by residual error's sensitivity defined in this paper, then removing the unimportant ones to produce a compact architecture of the neural network. In comparison to the original ELM, BELM, P-ELM, the SAP-ELM algorithm obtains competitive or improved generalization performance on some benchmark problems with more compact neural networks.

Acknowledgments. The research is financed by Natural Science Foundation of Jinan University (XKY1030).

References

1. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing (70), 489–501 (2006)
2. Huang, G.-B., Zhu, Q.-Y., Mao, K.Z., Siew, C.-K., Saratchandran, P., Sundararajan, N.: Can threshold networks be trained directly. IEEE Trans. Circuits Systems II 53(3), 187–191 (2006)
3. Huang, G.-B., Chen, L., Siew, C.-K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Networks 17(4), 879–892 (2006)
4. Huang, G.-B., Chen, L.: Enhanced random search based incremental extreme learning machine. Neurocomputing (71), 3460–3468 (2008)
5. Feng, G., Huang, G.-B., Lin, Q., Gay, R.: Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Networks 20(8), 1352–1357 (2009)
6. Teoh, E.J., Tan, K.C., Xiang, C.: Estimating the Number of Hidden Neurons in a Feedforward Network Using the Singular Value Decomposition. IEEE Trans Neural Networks 17(6), 1623–1629 (2006)
7. Huang, G., Babri, H.: Upper bounds on the number of hidden neu-rons in feedforward networks with arbitrary bounded nonlinear activa-tion functions. IEEE Trans Neural Networks 9(1), 224–229 (1998)
8. Huang, G.: Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans Neural Networks 14(2), 274–281 (2003)
9. Lecun, Y., Denker, J., Solla, S.: Optimal brain damage. Adv. Neural Inform Process. Syst. (2), 598–605 (1990)
10. Hassibi, B., Stork, D., Wolff, G.: Optimal brain surgeon and general network pruning. In: Proc. IEEE Int. Conf. Neural Networks, San Francisco, CA, vol. (1), pp. 293–299 (1992)
11. Zeng, X., Yeung, D.S.: Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure. Neurocomputing (69), 825–837 (2006)
12. Rong, H.-J., Ong, Y.-S., Tan, A.-H., Zhu, Z.: A fast pruned-extreme learning machine for classification problem. Neurocomputing (72), 359–366 (2008)
13. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: optimally pruned extreme learning machine. IEEE Trans Neural Networks 21(1), 158–162 (2010)
14. UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/`

Measuring Stability and Discrimination Power of Metrics in Information Retrieval Evaluation

Huaji Shi, Yanzhi Tan, Xiaolong Zhu, and Shengli Wu

School of Computer Science and Telecommunication Engineering
Jiangsu University, Zhenjiang, China, 212013

Abstract. Retrieval evaluation is always an important aspect in information retrieval (web search) and metrics are a key factor that needs to be carefully considered. In this paper, we propose a new method of measuring stability and discrimination power of a metric. The problem is initiated by Buckley and Voorhees. The advantage of the proposed method is that we are able to measure both aspects together in a systematic manner. Five metrics are tested in the study. They are average precision over all relevant documents, recall-level precision, normalized discount cumulative gain, precision at 10 documents level, and reciprocal rank. Experimental results show that normalized discount cumulative gain is the best, which is followed by average precision over all relevant documents, recall-level precision, precision at 10 documents level, while reciprocal rank is the worst.

1 Introduction

Retrieval evaluation is always an important aspect in information retrieval (web search). In order to evaluate a given retrieval model or a retrieval system, we need a test collection of documents, a set of topics, and relevance judgments for all the documents retrieved.

Retrieval evaluation is expensive mainly due to the fact that we need people to make judgment of documents' relevance. After some considerable effort on experimentation and testing for retrieval evaluation, we hope that the conclusions obtained are useful. In such a process, metrics need to be considered carefully due to their key role. Some major considerations include: the objective of the retrieval evaluation or which specific aspect of a retrieval result that we are looking for; whether any metric has different characteristics from the others and this should be considered in the evaluation design, for example, deciding that which metric or how many queries are going to be used.

Up to now, many different metrics have been proposed and quite a few of them have been used frequently. It is interesting to know the characteristics of those commonly used metrics. Among other aspects, stability and discrimination power are two important characteristics of any metric.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 8–15, 2013.

One paper published by Buckley and Voorhees [1] addresses the above-mentioned problem. Experiments are conducted using a historical TREC [1] data set to compare the stability and discrimination power of a group of metrics. In their methodology, for a given metric first they need to define a threshold, for example, 5%. Then they calculate the average performance of all the runs submitted to the same TREC task and count the number of pairs whose performance difference is above the given threshold. The tie rate is defined as the percentage of pairs whose performance difference is less than 5%. For those pairs whose performance difference is 5% or more, they calculate the error rates as follows: suppose there are two results A and B and A's average performance is better than B's average performance by 5% or more over all s topics, A is better than B by 5% or more for m topics, and B is better than A by 5% or more for n topics ($s \geq m + n$), then the error rate is defined as $n/(m + n)$. The explanation for the calculation is: for m times that the conclusion is the same between a topic and the average of all the topics, and for n times that the conclusion is opposite between a topic and the average of all the topics.

The characteristic of any specific metric can be partially represented by such tie rates and error rates. If we use the same data set to calculate tie rates and error rates for a group of different metrics, then we are in a fair position to compare the metrics involved. Usually lower error rates mean more stable metrics, and low tie rates mean more discriminative metrics. However, these two properties are somewhat contradictory. A metric performing well by one may perform poorly by the other. For example, in the experiment conducted by Buckley and Voorhees [1], recall at 1000 document-level has the lowest error rate, but its tie rate is higher than precision at 0.5 recall level, average precision, and recall-level precision. On the other hand, precision at 0.5 recall level has the lowest tie rate, but its error rate is higher than precision at 100 document level, average precision, recall-level precision, precision at 1000 document level, and recall at 1000 document level. In such situations, it is difficult to decide which metric is the best if we consider the two properties together.

The purpose of this paper is to find a better solution to address the above problem. We find that t test is a useful method. The experimental setting needed is exactly the same as that used by Buckley and Voorhees [1]. For a given threshold (significance level as we call it now), a given metric, and a group of results, instead of obtaining error rates and tie rates between all possible pairs, we calculate the percentage of pairs that the difference between the two results is significant at the given level. Thus only one value, instead of two, is calculated to indicate the goodness of the metric in a systematic way considering both stability and discrimination power.

[1] TREC (Text REtrieval Conference) is an information retrieval evaluation activity held annually since 1992. Its web site is located at `http://trec.nist.gov/`

2 Previous Work and Background Information

In information retrieval, retrieval evaluation is not a trivial problem. A lot of ranking-based metrics have been proposed and quite a few are used by researchers in their experiments and retrieval activities such as TREC, CLEF [2], etc.

In this study, five commonly used metrics are going to be tested. We discuss them here for convenience. Suppose that D is a collection of documents, q is a given query, and $total\_r$ is the total number of relevant documents in D for q. An information retrieval system ir provides a ranked list of documents $L = <d_1, d_2,..., d_n >$. Average precision over all relevant documents (or average precision, AP) is defined as

$$AP(L) = \frac{1}{total\_r} \sum_{i=1}^{total\_r} \frac{i}{t_i} \tag{1}$$

where t_i is the ranking position of the i-th relevant documents in the resultant list L. If some of the relevant documents in D do not appear in L, then we assume that they would appear in indefinite ranking positions. As an example, if there are 4 relevant documents in the whole collection and two of them are retrieved in the ranking positions of 2 and 4 in L for a given query, then AP(L) = 1/4*(1/2 + 2/4) = 0.25.

Recall-level precision (RP) is defined as the percentage of relevant documents in the top $total\_r$ documents in L.

Precision at 10 document level (P@10) is defined as the percentage of relevant documents in the top 10 documents in L.

Normalized discounted cumulative gain (NDCG, or NDCG@k) is introduced in [2]. Each ranking position in a resultant document list is assigned a given weight. The top ranked documents are assigned the heaviest weights since they are the most convenient ones for users to read. A logarithmic function-based weighting schema is proposed in [2], but first k (say, 2) documents are assigned the same weight of 1; then for any document ranked i that is greater than k, its weight is $w(i) = ln(k)/ln(i)$. Considering a resultant document list up to m documents, its discount cumulated gain (DCG, or DCG@k) is defined as

$$DCG(L, m) = \sum_{i=1}^{m} (w(i) * r(i))$$

where $r(i)$ is defined as: if the i-th document is relevant, then $r(i) = 1$; if the i-th document is irrelevant, then $r(i) = 0$. DCG can be normalized using a normalization coefficient DCG_best, which is the DCG value of the best resultant lists. Therefore, we have

$$NDCG(L, m) = \frac{1}{DCG\_best(L, m)} \sum_{i=1}^{t} (w(i) * r(i)) = \frac{DCG(L, m)}{DCG\_best(L, m)} \tag{2}$$

[2] Its web site is located at `http://www.clef-initiative.eu/`

The reciprocal rank (RR) is defined as

$$RR(L) = \frac{1}{f_r} \tag{3}$$

where f_r is the ranking position of the first relevant document that appears in the result L.

Many different metrics have been proposed and used in retrieval evaluation. Some papers are concerned with the performance of those metrics. Buckley and Voorhees [1] addresses the problem by considering stability and discrimination power of them. More specifically, for a group of results from different retrieval systems that work with the same collection of documents and the same set of queries, they compare the results using tie rates and error rates from all pairs of results so as to measure the stability and discrimination power of a given metric.

From another angle, Voorhees and Buckley [7] also use the error rate to predict the possible difference between two information retrieval systems in a new test condition, i.e., another randomly selected group of queries is going to be considered.

Lin and Hauptmann [3] provide a theoretical explanation to support the method proposed by Buckley and Voorhees [1]. Sakai also uses the same method that Buckley and Voorhees [1] propose to evaluate the goodness of a group of metrics under the condition of graded relevance judgment [6], in which relevance documents are divided into multiple levels to reflect how relevant a document may be to the information need. As an alternative, Sakai [5] uses the Bootstrap technique to evaluate the goodness of some selected metrics in retrieval evaluation.

More recently, Robertson and Kanoulas [4] investigate the variance of values for a group of metrics from query to query. Zhou et. al. [8] investigate how to evaluate aggregated results from vertical meta-search engines.

Zobel [9] finds that the t test is more reliable than the Wilcoxon and sign tests. This is why we use the t test, rather than some others, in this study.

3 Experimental Settings

For two samples with the same number of participants, t test can be used to compare the means of them:

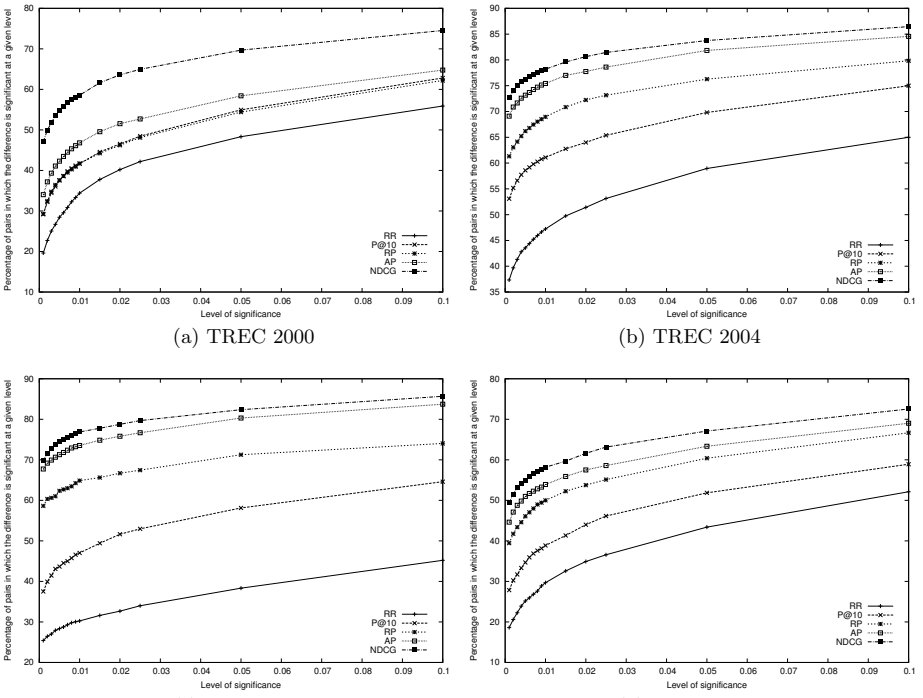
$$t = \frac{|X_1 - X_2|}{\sqrt{\frac{s_1^2 + s_2^2}{n}}} \tag{4}$$

Here X_1 and X_2 are the means, s_1^2 and s_2^2 are the unbiased estimators of the variances of the two samples, n is the number of participants in both samples. For a given significance level (e.g., 0.05), we can calculate t's value to see if the difference of the two means is significant at the given level. When calculating t, all the values in both samples matter.

Four data sets are used in this study. They are the results submitted to the TREC 2000 (TREC9) web track, the TREC 2004 robust track, the opinion task in TREC 2008 blog track and the TREC 2012 medical track. They are summarized in Table 1.

Table 1. Information of four groups of runs submitted to TREC

Group	Task	Total Number of runs	Number of selected runs	Number of queries
TREC 2000	web	105	89	50
TREC 2004	robust	110	109	249
TREC 2008	blog (opinion)	191	191	150
TREC 2012	medical	89	88	47



(a) TREC 2000 (b) TREC 2004

(c) TREC 2008 (d) TREC 2012

Fig. 1. Stability and discrimination power comparison of five metrics using four historical TREC data sets

In the TREC 2000 web track, quite a few runs submitted are very poor. We remove those whose performance measured by AP is below 0.05. Such a threshold is set arbitrarily. Because we think that those very poor results are not from proper information retrieval systems and it does not make much sense to include them in this study. In TREC 2004, one query (query 672) is dropped because it does not have any relevant documents; pircRB04d3 and pircRB04d5 are the same, so we keep one of them. In TREC 2012, 3 topics (topics 138, 159 and 166) are dropped because there are no relevant documents for them; one run (USFISDS1) is removed because its performance (measured by AP) is below 0.05.

Five metrics used in the study are AP (Average Precision over all relevant documents), RP (Recall-level Precision), NDCG (Normalized Discount Cumulative Gain), P@10 (Precision at 10 documents level), and RR (Reciprocal Rank).

4 Experimental Results

For all selected runs in a year group, we compare every possible pair of runs. For example, in TREC 2004, 109 runs are selected from all 110 runs submitted. There are 109*108/2 = 5886 pairs in total. For a given significance level (e.g., 0.05) and a given metric (e.g., AP), we see if the difference of performance (measured by AP) of the two is significant at the given level of 0.05. This needs to be looked at their performance at query level. Then a two-tailed t test can be carried out with all the query-level performance values to decide if the difference is significant at a given level. After calculations for all the pairs, we count the number of pairs in which the difference is significant. 15 significance levels are tested. They are 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.010, 0.015, 0.020, 0.025, 0.050, 0.100. The results are shown in Figures 1 for all 4 year groups.

From Figures 1, we can see that the curve of NDCG is always on the top, which is followed by AP, PR, P@10, and RR, in that order. There is only one exception. In TREC 2000, the curves of RP and P@10 are very close to each other. In general, such results are as expected. It confirms that NDCG is the best metric, which is followed by AP, RP, P@10, and RR, in that order, if we consider a metric's stability and discrimination power together. Further, comparison of any two metrics can be quantitative. For example, consider NDCG and RP in TREC 2004. the corresponding percentages are 83.76 and 76.28 for a significance level of 0.05. We may say that NDCG is better than RP by (83.76-76.28)/76.28 = 9.8% in such a setting.

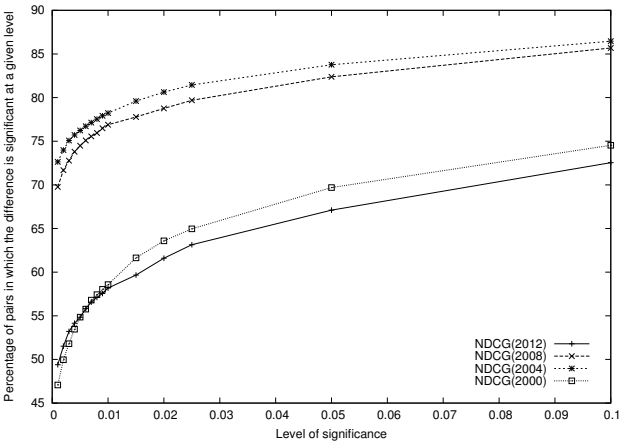


Fig. 2. The situation of metric NDCG in four different year groups

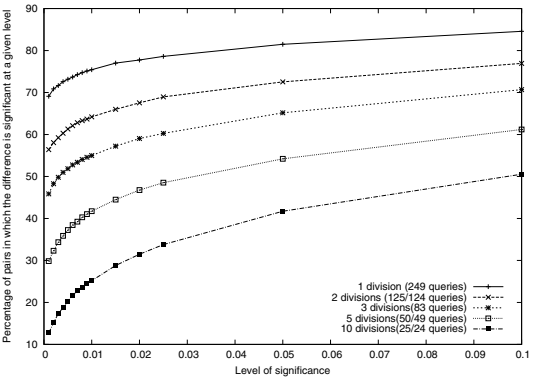


Fig. 3. Effect of query numbers on metric (AP) with the TREC 2008 data set

For any of the metrics involved, it performs very differently from one year group to another. TREC 2004 is the most favourable, TREC 2008 is in second place, while TREC 2000 and TREC 2012 are the worst. For clarity, Figure 2 shows the curves of NDCG. The situation is the same for the other 4 metrics. We hypothesize that the major impact factor is the number of queries in each year group, because TREC 2004 has the largest number (249), which is followed by TREC 2008 (150), while TREC 2000 and TREC 2012 only have 50 and 47 queries, respectively.

To test the above hypothesis, we carry out an experiment using the data set of TREC 2004 as follows: for a total number of 249 queries, we divide them into a few divisions equally. As an example, we may generate 3 equal divisions and each includes 83 queries consecutively. Then for each division, we measure each metric's stability and discriminant power as before using t test. Apart from 3, we also test the situation of 2, 5, and 10 divisions. When 2, 5, or 10 divisions are generated, one division has one query less than the other division or divisions. Otherwise, the same methodology is applied as in the case of 3 divisions. Figure 3 shows the results with metric AP. In Figure 3, each point is the average of all the divisions. We can observe that more divisions we generate, the lower the percentage of pairs we obtain whose difference is above a given significant level. The situation for other metrics is the same. Thus the hypothesis is confirmed.

5 Conclusions

In this paper, we have investigated a new method of evaluating metrics in information retrieval. The proposed method is reliable because it is based on the t test, which is a well-established statistical test and widely used for many problems in information retrieval and elsewhere. Both stability and discrimination power of a metric can be considered at the same time.

For the five metrics involved, we confirm that NDCG is the best, which is followed by AP, RP, P10, and RR, in that order. Comparison between any two metrics can also be quantitative, which has been shown in the experiment.

As of our future work, we plan to apply the same method to some selected metrics in the condition of graded relevance judgment.

References

1. Buckley, C., Voorhees, E.M.: Evaluating evaluation measure stability. In: Proceedings of ACM SIGIR Conference, Athens, Greece, pp. 33–40 (July 2000)
2. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 20(4), 442–446 (2002)
3. Lin, W., Hauptmann, A.: Revisiting the effect of topic set size on retrieval error. In: Proceedings of ACM SIGIR Conference, Salvador, Brazil, pp. 637–638 (August 2005)
4. Robertson, S., Kanoulas, E.: On per-topic variance in IR evaluation. In: Proceedings of ACM SIGIR Conference, Portland, USA, pp. 891–900 (August 2012)
5. Sakai, T.: Evaluating evaluation metrics based on the bootstrap. In: Proceedings of ACM SIGIR Conference, Seattle, USA, pp. 525–532 (August 2006)
6. Sakai, T.: On the reliability of information retrieval metrics based on graded relevance. Information Processing & Management 43(2), 531–548 (2007)
7. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: Proceedings of ACM SIGIR Conference, Tampere, Finland, pp. 316–323 (August 2002)
8. Zhou, K., Cummins, R., Lalmas, M., Jose, J.: Evaluating aggregated search pages. In: Proceedings of ACM SIGIR Conference, Portland, USA, pp. 115–124 (August 2012)
9. Zobel, J.: How reliable are the results of large-scale information retrieval experiments. In: Proceedings of ACM SIGIR Conference, Melbourne, Australia, pp. 307–314 (August 1998)

System for Monitoring and Optimization of Micro- and Nano-Machining Processes Using Intelligent Voice and Visual Communication

Dariusz Lipinski and Maciej Majewski

Koszalin University of Technology, Department of Mechanical Engineering
Raclawicka 15-17, 75-620 Koszalin, Poland
{dariusz.lipinski,maciej.majewski}@tu.koszalin.pl
http://kmp.wm.tu.koszalin.pl

Abstract. The article describes a new concept of voice and visual communication between the human operator and a system for monitoring and optimization of processes of micro- and nano-machining. The remote system for monitoring and optimization of process quality, which is equipped with a speech interface and artificial intelligence, is presented in exemplary application in the precision grinding process. The developed concept proposes an architecture of the system equipped with a data analysis layer, process supervision layer, decision layer, communication subsystem by speech and natural language, and visual communication subsystem using voice descriptions. In the proposed system, computational intelligence methods allow for real-time data analysis of monitored processes, configuration of the system, process supervision and optimization based on the process features and quality models. The concept allows for the development of universal and elastic systems which are independent of a type of manufacturing process, machining parameters and conditions. The developed block structure of the system allows for applications in monitoring of many other processes of micro- and nano-machining.

Keywords: monitoring and optimization, micro- and nano-machining processes, intelligent system, intelligent interface, process quality, measurement data analysis, modern machining process, artificial intelligence.

1 Introduction

Processes of micro- and nano-machining in the industry of today can be performed using a hybrid system for monitoring, optimization and forecasting of machining process quality, equipped with a layer of distance voice and visual communication between the system and human operators. The system of monitoring and optimization of the quality of the processes of micro- and nano-machining, presented in exemplary application in the precision grinding processes, features the possibility for many other applications, future development and experiments. Its main tasks include: modeling of the process, assessment of inaccuracy effects, identification of inaccuracy causes, optimization of the process conditions and parameters.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 16–23, 2013.

The scientific aim of the research is to develop complex fundamentals of building new systems (fig. 1) for monitoring and optimization of micro- and nano-machining processes using intelligent voice and visual communication. The

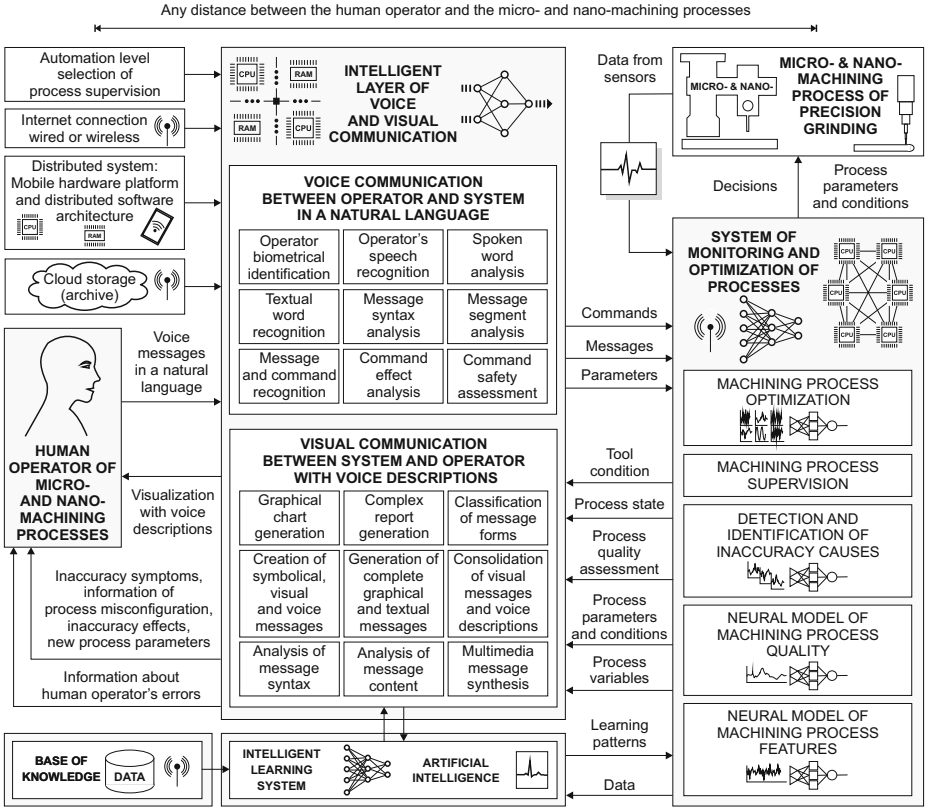


Fig. 1. Concept of voice and visual communication between the operator and a system for monitoring and optimization of micro- and nano-machining processes.

design and implementation of intelligent systems for monitoring and optimization of manufacturing processes is an important field of research. In these systems, a natural language interface using speech is ideal because it is the most natural, flexible, efficient, and economical form of human communication. This concept proposes a novel approach to intelligent systems for monitoring and optimization of the processes of micro- and nano-machining using intelligent voice and visual communication, with particular emphasis on their ability to be truly flexible, adaptive, human error-tolerant, and supportive both of human operators and intelligent agents in distributed systems architectures.

Application of intelligent interactive systems for monitoring and optimization of machining processes using a natural language offers many advantages.

It ensures robustness against human operator errors and efficient supervision of manufacturing processes with adjustable level of automated supervision. Natural language interfaces also improve the cooperation between a human operator and a system in respect to the richness of communication. Further, intelligent interaction allows for higher organization level of manufacturing processes, which is significant for their efficiency and humanization. Decision and optimization systems can be remote elements of manufacturing processes.

The design of the proposed system can be considered as an attempt to create a standard intelligent system for monitoring and optimization of machining processes using natural language communication. It is very significant for the development of new effective and flexible manufacturing methods. It can also contribute for increase of efficiency and decrease of costs of the production processes. This system provides an innovative solution allowing for more complete advantages of modern manufacturing processes nowadays.

2 The State of the Art

There is a need for remote systems of monitoring and optimization of machining processes in reconfigurable manufacturing systems to reduce bottlenecks that occur in associated tasks to be performed by these systems using technological devices. The tasks include: modeling of the process features and quality, assessment of inaccuracy effects, identification of inaccuracy causes, optimization of the process conditions and parameters. These bottlenecks can occur as a result of the mass production of custom products.

The sustainability of existing manufacturing resources and enhancement of the machining efficiency is an important field of applied research. The current research and recent advances in development of prototypes of systems for monitoring and optimization of machining processes are described in articles [1,2,3,4]. Those systems consist of computational algorithms for sensor base monitoring and control, and simulations of virtual machining processes. Those systems have been accepted for real-time decision making, sustainable development and efficient use of machining resources.

In many potential applications of these monitoring and optimization systems, the limiting factor may be an ability of the system to model the process features and quality, process the inaccuracy symptoms, solve the misconfiguration issues, compensate the inaccuracy effects, determine the process conditions and parameters. The integration of these systems and remote communication in applications faces a challenge of accuracy, robustness and portability. The integration also gives rise to some new challenges which include integration strategies, coordination of the components with system outputs, and adaptation towards reconfigurable manufacturing systems.

This article offers an approach by using the developed concept of the system of monitoring and optimization of the processes of micro- and nano-machining to deal with the above problems. Selected research article [5] presents innovative solutions in supervision of precise grinding processes and development of a hybrid

system for monitoring, optimization and forecasting of machining process quality. Articles [6,7,8,9,10,11] describe the developed solutions in the range of intelligent voice communication between human operators and technical devices.

3 Description of the System

The proposed system of monitoring and optimization of the processes of micro- and nano-machining, presented in exemplary application in the precision grinding processes, features the universality of application of the developed intelligent algorithms and possibility of application in many other processes of micro- and nano-machining in manufacturing. The developed block structure of the system allows for the development of universal and elastic systems which are independent of a type of manufacturing process, machining parameters and conditions.

The developed concept proposes an architecture of the system (fig. 2) equipped with a data analysis layer, process supervision layer, decision layer, communication subsystem by speech and natural language, and visual communication subsystem using voice descriptions.

The novelty of the system consists of inclusion of adaptive intelligent layers for data analysis, supervision and decision. The system is also capable of analysis of the supervised machining process, configuration of the supervision system, neural modeling of process features, neural modeling of process quality, detection of the inaccuracies, estimation of the inaccuracies results, compensation of the inaccuracies results, and selection of the machining parameters and conditions. In addition it makes an assessment of the correctness of the operator's decisions. The system also consists of mechanisms for meaning analyses of operator's messages and commands given by voice in a natural language, and various visual communication forms with the operator using voice descriptions.

The interaction between the operator and the system by speech and natural language contains intelligent mechanisms for operator biometric identification, speech recognition, word recognition, recognition of messages and commands, syntax analysis of messages, and safety assessment of commands. The interaction between the system and the operator using visual messages with voice descriptions includes intelligent mechanisms for generation of graphical and textual reports, classification of message forms, generation of messages in the graphical and textual forms, consolidation and analysis of message contents, synthesis of multimedia messages.

In the system, the data output layer provides the data and information about the machining parameters and conditions, tool condition, process state, process quality, and process variables. The data input layer contains commands for system control, commands for process monitoring and optimization, commands for process supervision, messages for configuration and interaction with the system of monitoring and optimization.

The structure of the system for monitoring and optimization of micro- and nano-machining processes is presented in abbreviated form on Fig. 3. The numbers in the cycle represent the successive phases of information processing. The

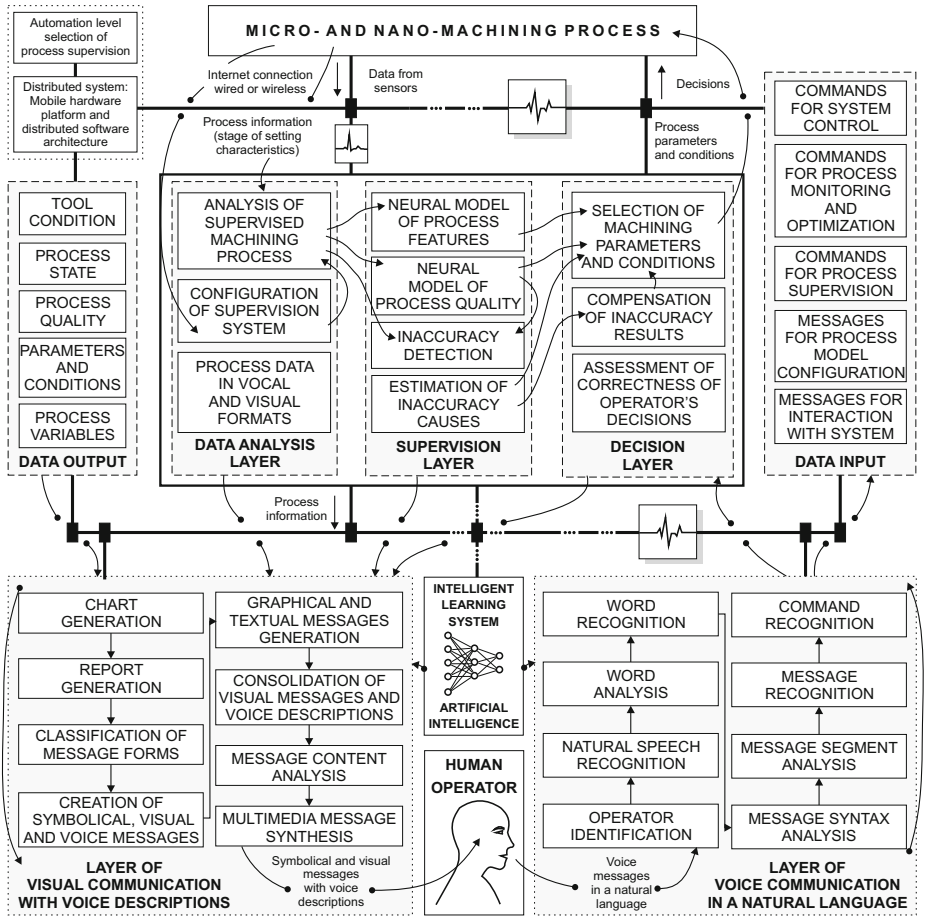


Fig. 2. Concept of a system of distance monitoring and optimization of micro- and nano-machining processes using voice and visual communication

system performs biometric identification of the human operator whose spoken messages in a natural language are converted to text and numerical values. The recognized text is processed by the meaning analysis subsystem using artificial intelligence methods. The processing involves meaning analysis of words, messages and commands in a natural language. The results are recognized meaningful messages and commands, which are sent to the subsystem of data analysis.

The results after the analysis of the supervised process and configuration of the process supervision are further processed with the subsystem of supervision of micro- and nano-machining processes, which is composed of several specialized modules. The subsystem contains modules for modeling of the process features, modeling of the process quality, optimization of the process, determination of allowable changes of the process features, assessment of the grade of changes of the features, detection of inaccuracy causes, identification of inaccuracy causes,

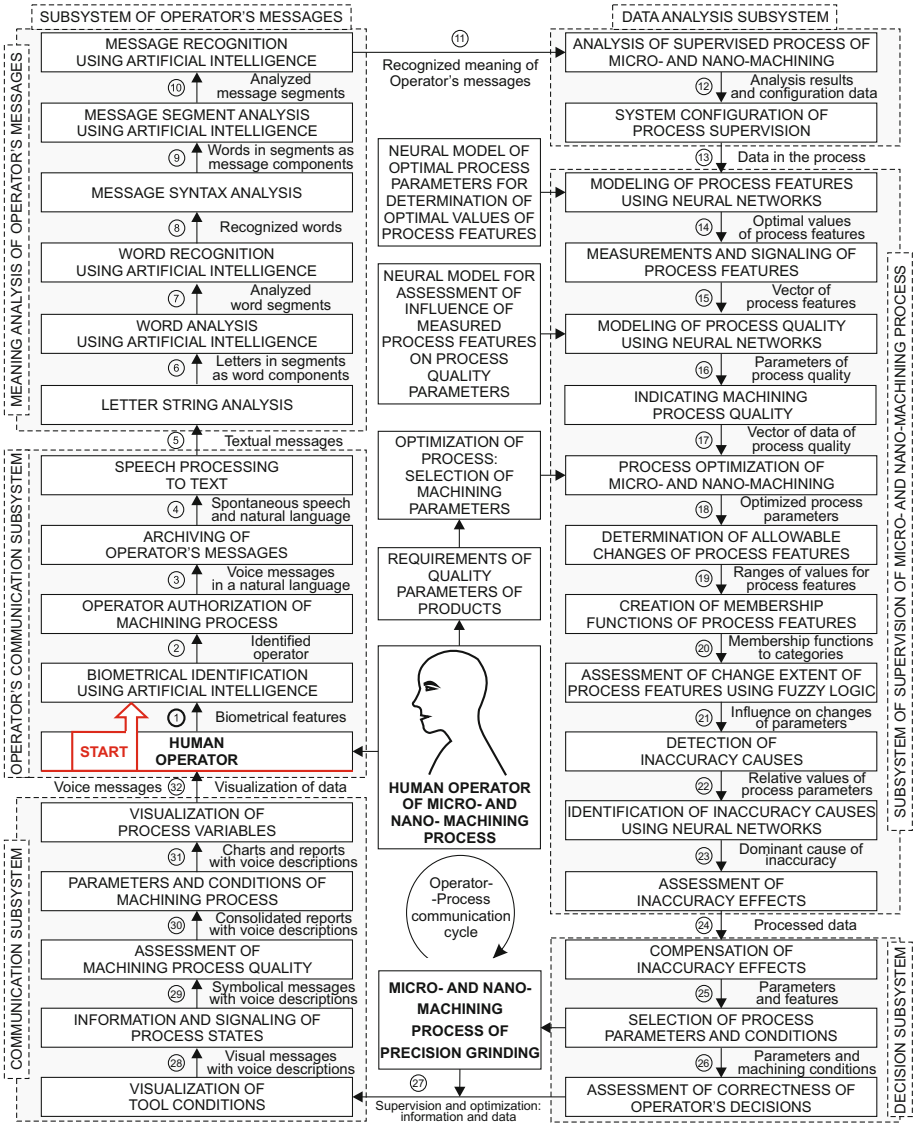


Fig. 3. Structure of a system of distance monitoring and optimization of the machining process using voice and visual communication between the operator and the system

and assessment of inaccuracy effects. The results are subsequently sent to the decision subsystem for compensation of inaccuracy effects, selection of the process parameters and conditions, assessment of correctness of the operator's decisions. The next phase is processing in the subsystem of communication for visualization of the tool condition, signaling of the process states, reporting the assessment results of the machining process quality, reporting the parameters and conditions of the process, visualizations of the variables.

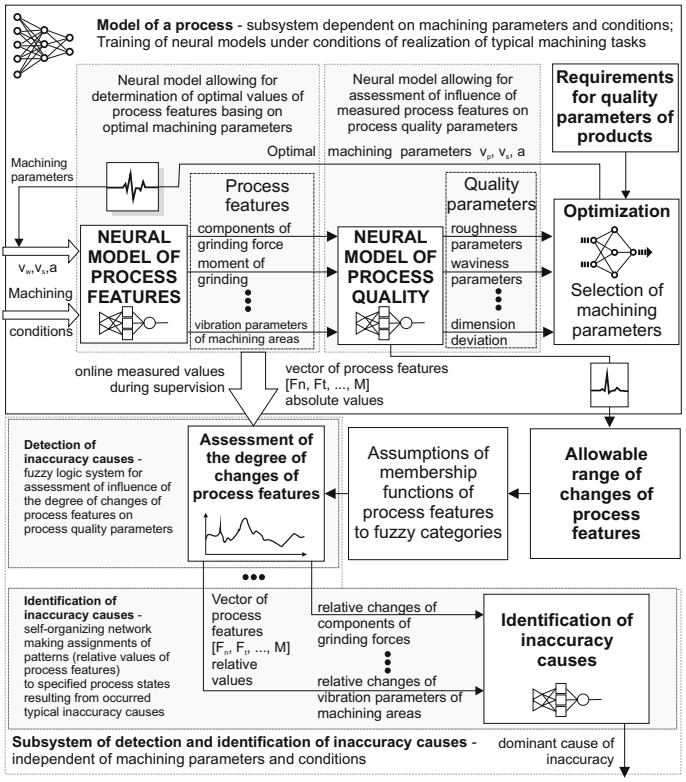


Fig. 4. System for detection of inaccuracies and optimization of machining parameters

The system is equipped with the neural model of the optimal process parameters for determination of optimal values of the process features and the neural model for assessment of influence of the measured process features on the process quality parameters. The models are used in a system (fig. 4) for detection of inaccuracies and optimization of machining parameters.

4 Conclusions and Perspectives

The proposed concept of the system for monitoring and optimization of processes of micro- and nano-machining, equipped with a speech interface and artificial intelligence, allows for the development of universal and elastic systems which are independent of a type of manufacturing process, machining parameters and conditions. The developed block structure of the system allow for application in monitoring of many other processes of micro- and nano-machining. The condition of effectiveness of the system is to equipped it with intelligent mechanisms for modeling of the process features and quality, assessment of inaccuracy effects, identification of inaccuracy causes, optimization of the process conditions and parameters. The experimental results of the proposed system show its promising

performance. The concept can be used for further development and experiments. The system is both effective and flexible which makes its applications possible. As an interface, it allows for more robustness to human's errors. The proposed complex solution also eliminates scarcities of the typical co-operation between human operators and technological devices.

Acknowledgements. The authors would like to thank The National Science Center in Poland (NCN) who provided financial support for this work.

References

1. Abellan-Nebot, J.V., Subiron, R.F.: A Review of Machining Monitoring Systems Based on Artificial Intelligence Process Models. International Journal of Advanced Manufacturing Technology 47, 237–257 (2010)
2. Brown, J.A.: Modern Manufacturing Processes. Industrial Press, New York (2011)
3. Teti, R., Jemielniak, K., O'Donnell, G., Dornfeld, D.: Advanced Monitoring of Machining Operations. CIRP Annals - Manufacturing Technology 59, 717–739 (2010)
4. Groover, M.P.: Fundamentals of Modern Manufacturing. Wiley & Sons, Inc. (2007)
5. Lipiński, D., Kacalak, W.: Assessment of the Accuracy of the Process of Ceramics Grinding with the Use of Fuzzy Interference. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 596–603. Springer, Heidelberg (2007)
6. Kacalak, W., Majewski, M.: New Intelligent Interactive Automated Systems for Design of Machine Elements and Assemblies. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part IV. LNCS, vol. 7666, pp. 115–122. Springer, Heidelberg (2012)
7. Kacalak, W., Majewski, M.: Effective Handwriting Recognition System Using Geometrical Character Analysis Algorithms. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part IV. LNCS, vol. 7666, pp. 248–255. Springer, Heidelberg (2012)
8. Kacalak, W., Majewski, M.: Natural Language Human-Robot Interface Using Evolvable Fuzzy Neural Networks for Mobile Technology. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS, vol. 5754, pp. 480–489. Springer, Heidelberg (2009)
9. Majewski, M., Zurada, J.M.: Sentence Recognition Using Artificial Neural Networks. Knowledge-Based Systems 21(7), 629–635 (2008)
10. Majewski, M., Kacalak, W.: Intelligent System for Natural Language Processing. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 742–747. Springer, Heidelberg (2006)
11. Stuart, K.D., Majewski, M., Trelis, A.B.: Intelligent Semantic-Based System for Corpus Analysis through Hybrid Probabilistic Neural Networks. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (eds.) ISNN 2011, Part I. LNCS, vol. 6675, pp. 83–92. Springer, Heidelberg (2011)

Racing for Unbalanced Methods Selection

Andrea Dal Pozzolo[1], Olivier Caelen[2],
Serge Waterschoot[2], and Gianluca Bontempi[1]

[1] Machine Learning Group, Computer Science Department, Faculty of Sciences ULB, Université Libre de Bruxelles, Brussels, Belgium
[2] Business Analytics Competence Center, Atos Worldline, Belgium

Abstract. State-of-the-art classification algorithms suffer when the data is skewed towards one class. This led to the development of a number of techniques to cope with unbalanced data. However, as confirmed by our experimental comparison, no technique appears to work consistently better in all conditions. We propose to use a racing method to select adaptively the most appropriate strategy for a given unbalanced task. The results show that racing is able to adapt the choice of the strategy to the specific nature of the unbalanced problem and to select rapidly the most appropriate strategy without compromising the accuracy.

Keywords: Unbalanced data, Fraud detection, Racing.

1 Introduction

Learning from unbalanced datasets is a difficult task since most learning algorithms are not designed to cope with a large difference between the number of cases belonging to different classes [2]. The unbalanced nature of the data is typical of many applications such as medical diagnosis, text classification and oil spills detection. Credit card fraud detection [16,6,17] is another well-known instance of highly unbalanced problem since (fortunately) the number of fraudulent transactions is typically much smaller than legitimate ones. In literature several methods for dealing with unbalanced datasets have been proposed. They essentially belong to the following categories: sampling, ensemble and distance-based.

The ratio between majority and minority class is not the only factor that determines the difficulty of a classification/detection task. Another influential factor is the amount of overlapping of the classes of interest [9]. Other studies [10,14] showed that some methods are superior to others under certain conditions.

All these supports the idea that under different conditions, such as different datasets and algorithms, the best methods may change. In particular, in credit card fraud detection, the fraudulent behaviour evolves over the time changing the distribution of the frauds and a method that worked well in the past could become inaccurate afterward. Since in real large variate tasks it is hard to know a priori the nature of the unbalanced tasks, the user is recommended to test all

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 24–31, 2013.

techniques with a consequent high computational cost. In this context we propose a racing strategy to accelerate the search of the strategy in an unbalanced problem.

In this paper we first review the most common methods for dealing with unbalanced data in a supervised context. Then we make an exhaustive comparison of these methods on a real credit-card fraud dataset and nine public benchmark datasets. The results show that there is no balancing technique which is consistently the best one and that the best method depends on the algorithm applied as well as the the dataset used. For this reason, we propose the adoption of a racing strategy [15] to automatically select the most adequate technique for a given dataset. The rationale of the racing strategy consists in testing in parallel a set of alternative balancing strategies on a subset of the dataset and to remove progressively the alternatives which are significantly worse. Our results show that by adopting a racing strategy we are able to select in an efficient manner either the best balancing method or a method which is not significantly different from the best one. Moreover racing is able to reduce consistently the computation needed before finding the right methods for the dataset.

2 Strategies for Unbalanced Classification

Let us consider a binary classification task where the distribution of the target class is highly skewed. Let us call the majority class negative (coded as 0) and the minority class as positive (coded as 1). When the data is unbalanced, standard machine learning algorithms tend to be overwhelmed by the majority class [10]. There are several methods that deal with this problem and we can distinguish them into the following main categories: sampling, ensemble, distance-based and hybrid.

Sampling techniques do not take into consideration any class information in removing or adding observations, yet they are easy to implement and to understand. *Undersampling* [7] consists in down-sizing the majority class by removing observations at random until the dataset is balanced. In an unbalanced problem it is realistic to assume that many observations of the majority class are redundant and that by removing some of them at random the resulting distribution should not change much. *Oversampling* [7] consists in up-sizing the small class at random decreasing the level of class imbalance. By replicating the minority class until the two classes have equal frequency, oversampling increases the risk of overfitting [7] by biasing the model towards the minority class. *SMOTE* [5] over-samples the minority class by generating synthetic minority examples in the neighborhood of observed ones. The idea is to form new minority examples by interpolating between examples of the same class. This has the effect of creating clusters around each minority observation.

Ensemble methods combine an unbalanced method with a classifier to explore the majority and minority class distribution. *BalanceCascade* [13] is a supervised strategy to undersample the majority class. This method iteratively removes the majority class instances that are correctly classified by a boosting algorithm.

EasyEnsemble [13] learns different aspects of the original majority class in an unsupervised manner. This is done by creating different balanced training sets by *Undersampling*, learning a model for each dataset and then combining all predictions as in bagging.

The following methods make use of distance measures between input points either to undersample or to remove noisy and borderline examples of each class. *Tomek link* [19] removes observations from the negative class that are close to the positive region in order to return a dataset that presents a better separation between the two classes. Let us consider two input examples x_i and x_j belonging to different classes, and let $d(x_i, x_j)$ be their distance. A (x_i, x_j) pair is called a *Tomek link* if there is no example x_k, such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. Negative examples that are Tomek links are then removed reducing the majority class. *Condensed Nearest Neighbor (CNN)* [8] is used to select a subset S from the original unbalanced set T which is consistent with T in the sense that S classifies T correctly with the one-nearest neighbor rule. Since noisy examples are likely to be misclassified, many of them will be added to the S set which means CNN rule is extremely sensitive to noise [21]. *One-sided Selection* (OSS) [11] is an undersampling method resulting from the application of Tomek links followed by the application of CNN. *Edited Nearest Neighbor (ENN)* [20] removes any example whose class label differs from the class of at least two of its three nearest neighbors. In this way majority examples that fall in the minority region and isolated minority examples are removed. *Neighborhood Cleaning Rule (NCL)* [12] modifies the ENN method by increasing the role of data cleaning. Firstly, NCL removes negatives examples which are misclassified by their 3-nearest neighbors. Secondly, the neighbors of each positive examples are found and the ones belonging to the majority class are removed.

A set of hybrid strategies can be easily created by combining sampling, ensemble and distance base techniques. In particular we consider the following hybrids: *SMOTEnsemble* (EasyEnsemble with SMOTE), *EnnSmote* (SMOTE after ENN), *TomekUnder* (Undersampling after Tomek), *TomekEasyEnsemble* (EasyEnsemble after Tomek) and *TomekSMOTE*: (SMOTE after Tomek).

3 Racing for Strategy Selection

The variety of approaches discussed in Section 2 suggests that in a real situation where we have no prior information about the data distribution, it is difficult to decide which unbalanced strategy to use. In this case testing all alternatives is not an option either because of the associated computational cost.

A possible solution comes form the adoption of the Racing approach which was proposed in [15] to perform efficiently model selection in a learning task. The principle of Racing consists in testing in parallel a set of alternatives and using a statistical test to determine if an alternative is significantly worse than the others. In that case such alternative is discarded from the competition, and the computational effort is devoted to differentiate the remaining ones. Historically the first example of Racing method is called Hoeffding Race since it relies on

the Hoeffding theorem to decide when a model is significantly worse than the others. The *F-race* version was proposed in [4] and combines the Friedman test with Hoeffding Races [15] to eliminate inferior candidates as soon as enough statistical evidence arises against them. In F-race, the Friedman test is used to check whether there is evidence that at least one of the candidates is significantly different from others and post-tests are applied to eliminate those candidate that are significantly worse than the best one.

Here we adopt F-Race to search efficiently for the best strategy for unbalanced data. The candidates are assessed on different subsets of data and, each time a new assessment is made, the Friedman multiple test is used to dismiss significantly inferior candidates. We used a 10 fold cross validation to provide the assessment measure to the race. If a candidate is significantly better than all the others than the race is terminated without the need of using the whole dataset. In case there is not evidence of worse/better methods, the race terminates when the entire dataset is explored and the best candidate is the one with the best average result.

4 Experimental Results

We tested the 15 strategies for unbalanced data discussed in Section 2 on the datasets of the following table:

Dataset ID	Dataset name	Size	Input	Prop 1	Class 1
1	breastcancer	698	10	34.52%	class =4
2	car	1727	6	3.76%	class = Vgood
3	forest	38501	54	7.13%	class = Cottonwood/Willow
4	letter	19999	16	3.76%	letter = W
5	nursery	12959	8	2.53%	class = very_recom
6	pima	768	8	34.89%	class = 1
7	satimage	6433	36	9.73%	class = 4
8	women	1472	9	22.62%	class = long-term
9	spam	4601	57	42.14%	class =1
10	fraud	527026	51	0.39%	Fraud = 1

The first 9 datasets are from UCI [1] repository. UCI datasets that have originally more than two response classes are transformed into binary by picking one class as the minority and joining all the others to form the majority class.

The credit card fraud dataset was provided by a payment service provider in Belgium. The fraud dataset is not available to the public because of the confidentiality of the data. In fraud detection it is important to have a high Precision and Recall, therefore we used F-measure as performance metric since it is able to combines Precision and Recall into a single metric.

We started by testing on the fraud dataset different supervised algorithms such as Random Forest, Neural Network, Support Vector Machine and Naive Bayes. For the sake of reproducibility we used the implementation provided in the R software [18] with default parameters.

Each algorithm was first tested on the entire fraud dataset using a 10 fold cross validation for all the strategies. Naive Bayes is the only algorithm whose performance is not sensitive to the adopted strategy and comparable to the unbalanced

strategy (i.e. the strategy leaving the data in the original status). In Figure 1 we can notice that for Support Vector Machine and Random Forest the group of techniques that include oversampling (i.e. Oversampling and SMOTE-related strategies) are performing better than the others. For most of the algorithms, distance-based strategy (ENN, NCL, CNN, OSS and Tomek) perform as bad as the unbalanced case. The highest F-measure is reached using Random Forest algorithm with SMOTEnsemble.

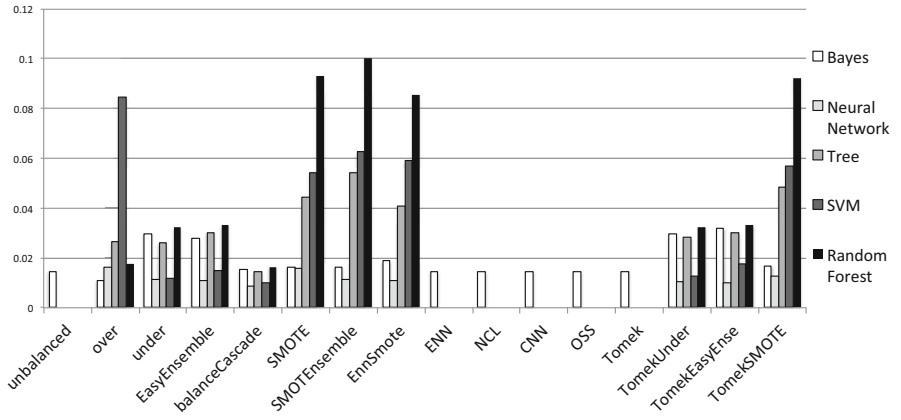


Fig. 1. Comparison of strategies for unbalanced data for the Fraud dataset in terms of F-measure (the higher the better)

At this point we tested our strategies for unbalanced data on the public UCI datasets. On some of them (e.g. Breast Cancer) no significant differences between strategies was detected.

We used the Friedman test to detect differences in the methods across all datasets. A post-hoc test based upon paired t-test of the ranks was used to decide which methods are significantly different from each other (Figure 2). From Figure 2 we can notice that with Random Forest, SMOTEnsemble was statistically better than many of unbalanced strategies, while oversampling was the best for SVM.

What emerges from this study is that there is no single strategy which is coherently superior to all others in all conditions (i.e. algorithm and dataset). Even if sometimes it is possible to find a strategy that is statistically better than others it is computationally demanding testing all strategies in several dataset and algorithms.

We decided to adopt the F-race algorithm implemented in [3] (with default parameters) to automatise the way to select the best strategy for unbalanced data. In Table 1 we used the F-race method to automatically select the unbalanced strategy.

Let us remark that for almost all datasets F-race is able to return the best method according to the cross validation (CV) assessment. In the case of Pima

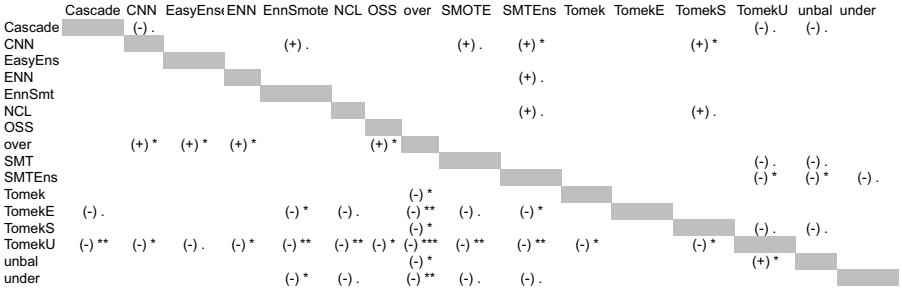


Fig. 2. Comparison of strategies using a post-hoc Friedman test in terms of F-measure over multiple datasets. RF results are above the diagonal, SVM below. A cell above the diagonal is marked as (+) if the rank difference between the method in the column and the one in the row is significantively positive, (-) otherwise. Below, (+) denotes that the method in the row is significantively better than the one in the column, (-) otherwise. The matrix contains the level of significance of the t-test annotated as follows: ***, **, * and . for $\alpha = 0.001, 0.01, 0.05, 0.1$. A cell is left empty if the test is not significant.

Table 1. Comparison of Cross Validation and F-race results with Random Forest and Support Vector Machines in terms of F-measure

Dataset	Algo	Exploration	Method	N test	Gain	Mean	Sd	Pval
Fraud	RF	best CV	SMOTEEnsemble	160	-	0.100	0.016	-
		F-race	SMOTEEnsemble	44	73%			
	SVM	best CV	over	160	-	0.084	0.017	-
		F-race	over	46	71%			
Breast Cancer	RF	best CV	balanceCascade	160	-	0.963	0.035	-
		F-race	balanceCascade	160	0%			
	SVM	best CV	under	160	-	0.957	0.038	-
		F-race	under	160	0%			
Car	RF	best CV	OSS	160	-	0.970	0.039	-
		F-race	OSS	108	33%			
	SVM	best CV	over	160	-	0.944	0.052	-
		F-race	over	93	42%			
Forest	RF	best CV	balanceCascade	160	-	0.911	0.012	-
		F-race	balanceCascade	60	63%			
	SVM	best CV	ENN	160	-	0.809	0.011	-
		F-race	ENN	64	60%			
Letter	RF	best CV	balanceCascade	160	-	0.981	0.010	-
		F-race	balanceCascade	73	54%			
	SVM	best CV	over	160	-	0.953	0.022	-
		F-race	over	44	73%			
Nursery	RF	best CV	SMOTE	160	-	0.809	0.047	-
		F-race	SMOTE	76	53%			
	SVM	best CV	over	160	-	0.875	0.052	-
		F-race	over	58	64%			
Pima	RF	best CV	under	160	-	0.691	0.045	-
		F-race	under	136	15%			
	SVM	best CV	EasyEnsemble	160	-	0.675	0.071	0.107063
		F-race	TomekUnder	110	31%	0.672	0.067	
Satimage	RF	best CV	balanceCascade	160	-	0.719	0.033	-
		F-race	balanceCascade	132	18%			
	SVM	best CV	balanceCascade	160	-	0.662	0.044	-
		F-race	balanceCascade	90	44%			
Spam	RF	best CV	SMOTE	160	-	0.942	0.015	-
		F-race	SMOTE	122	24%			
	SVM	best CV	SMOTEEnsemble	160	-	0.917	0.018	0.266028
		F-race	SMOTE	135	16%	0.9178762	0.02	
Women	RF	best CV	TomekUnder	160	-	0.488	0.051	-
		F-race	TomekUnder	150	6%			
	SVM	best CV	EnnSmote	160	-	0.492	0.073	-
		F-race	EnnSmote	102	36%			

and Spam datasets, F-race returns a sub-optimal strategy whose accuracy is however not significantly different from the best one (Pvalue greater than 0.05).

The main advantage of Racing is that bad methods are not tested on the whole dataset reducing the computation needed. Taking into consideration the 15 methods and the unbalanced case, in a 10 fold cross validation we have 160 tests to make (10 folds x 16 methods). In the case of F-race the number of total tests depends upon how many folds are needed before F-race finds the best method. The *Gain* column of Table 1 shows the computational gain (in percentage of the the CV tests) obtained by using F-race. Apart from the Breast Cancer dataset in all the other cases F-race allows a significant computational saving with no loss in performance.

5 Conclusion

Recent literature in data mining and machine learning is plenty of research works on strategies to deal with unbalanced data. However a definitive answer on the best strategy to adopt is yet to come. Our experimental results support the idea that the final performance is extremely dependent on the data nature and distribution.

This consideration has lead us to adopt the F-race strategy where different candidates (unbalanced methods) are tested simultaneously. We have showed that this algorithm is able to select few candidates that perform better than other without exploring the whole dataset. F-race was able to get results similar to the cross validation for most of the dataset.

As far as the fraud dataset is concerned, we found SMOTEnsemble together with RandomForest to be the best strategy. In this dataset the unbalanced strategy chosen had a big impact on the accuracy of the results. However, as the frauds evolve over the time the same method could become sub-optimal in the future. In this context the F-race contribution to the selection of the best strategy is crucial in order to have a detection system that adapts quickly to the new data distribution. Within the UCI datasets we noticed that some tasks are much easier (high accuracy) than the others and they may not have an unbalanced method that performs significantly better than the others.

Acknowledgment. The work of Andrea Dal Pozzolo was supported by the Doctiris programe of Innoviris (Brussels Institute for Research and Innovation), Belgium.

References

1. D.N.A. Asuncion. UCI machine learning repository (2007)
2. Batista, G., Carvalho, A., Monard, M.: Applying one-sided selection to unbalanced datasets. In: Cairó, O., Cantú, F.J. (eds.) MICAI 2000. LNCS, vol. 1793, pp. 315–325. Springer, Heidelberg (2000)

3. Birattari, M.: Race: Racing methods for the selection of the best, R package version 0.1.59 (2012)
4. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 11–18 (2002)
5. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: synthetic minority oversampling technique. Arxiv preprint arXiv:1106.1813 (2011)
6. Clark, P., Niblett, T.: The cn2 induction algorithm. Machine Learning 3(4), 261–283 (1989)
7. Drummond, C., Holte, R., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II. Citeseer (2003)
8. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory (1968)
9. Holte, R.C., Acker, L.E., Porter, B.W., et al.: Concept learning and the problem of small disjuncts. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, vol. 1. Citeseer (1989)
10. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis 6(5), 429–449 (2002)
11. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: Machine Learning-International Workshop Then Conference, pp. 179–186. Morgan Kaufmann Publishers, Inc. (1997)
12. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 63–66. Springer, Heidelberg (2001)
13. Liu, X., Wu, J., Zhou, Z.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(2), 539–550 (2009)
14. Lin, K.T.M., Yao, X.: A dynamic sampling approach to training neural networks for multi-class imbalance classification. IEEE Transactions on Neural Networks and Learning Systems 24, 647–660 (2013)
15. Maron, O., Moore, A.: Hoeffding races: Accelerating model selection search for classification and function approximation, p. 263. Robotics Institute (1993)
16. Olshen, L., Stone, C.: Classification and regression trees. Wadsworth International Group (1984)
17. Quinlan, J.R.: C4. 5: programs for machine learning, vol. 1. Morgan Kaufmann (1993)
18. R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2011) ISBN 3-900051-07-0
19. Tomek, I.: Two modifications of cnn. IEEE Trans. Syst. Man Cybern. 6, 769–772 (1976)
20. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics (3), 408–421 (1972)
21. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)

Super-Resolution from One Single Low-Resolution Image Based on R-KSVD and Example-Based Algorithm

FangMei Fu[1] , Jiong Jia[1,*], ZhongLong Zheng[1,2,*],
Fan Yang[1], Li Guo[1], HaiXin Zhang[1], and MuDan Yu[1]

[1] Department of Computer Science, Zhejiang Normal University,
321004 Zhejiang, China
[2] Department of Computer Science, University of California, Merced
95340 California, USA
jia@zjnu.cn, zzheng@ucmerced.edu

Abstract. With the development of sparse coding and compressive sensing, image Super-resolution (SR) reconstruction attracts extensive attentions. In this paper, we mainly focus on recovering super-resolution version given only one single low-resolution (LR) image. The proposed method is combined with the example-based algorithm, which also exploits the relationship between the low image patches and the high image patches. Firstly, the proposed method applies guided filter, the first-order and second-order derivatives to extract multiple features from LR images, which superior to using only one feature space. Then, the effective dictionary is constructed by a novel algorithm called Relaxation K-SVD (R-KSVD). R-KSVD relaxes the constraints of Orthogonal Matching Pursuit method (R-OMP) in training dictionary for K-SVD algorithm. Finally, a new approach is presented to estimating better HR residual image in the Back Projection. Experimental results demonstrate the superiority of our algorithm in both visual fidelity and numerical measures.

Keywords: Super-resolution, R-KSVD, sparse representation, guided filter.

1 Introduction

Image super-resolution (SR) is currently an increasing active area of research in image processing. The SR problem aims to recover a high-resolution (HR) image from one or more low-resolution (LR) images. It is an inverse problem only under reasonable assumptions and prior knowledge conditions. Existing SR reconstruction algorithms [1-4] mainly include reconstruct-based methods and example-based methods. Reconstruct-based methods degrade rapidly if the magnification factor is too large or if there are not enough LR inputs to constrain the solution [1]. Example-based methods alleviate the difficulty above, which generate a dictionary or other mapping form to identify the relationship between LR and HR images. Then the corresponding HR images can be reconstructed via this relationship.

* Corresponding authors.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 32–39, 2013.

However, in terms of SR problem from only one single LR image, those traditional methods demonstrate poor performances. To alleviate it, numerous single-image super-resolution algorithms [6], [13], [14] are proposed, which generate a visually pleasing HR image only from a given LR input image. Comparing with the traditional example-based algorithm, those algorithms learn the dictionary based on the image self-similarity property. They avoid the influence of extra database while maintaining comparable super-resolution image quality.

Motivated by Glasner [6] method, we propose a novel SR approach. At first, we render the dataset by the input LR image according to the self-similarity property. Then, the effective dictionary is constructed by a novel algorithm called Relaxation K-SVD (R-KSVD). Furthermore, in order to obtain better HR residual image, we improve the Back Projection algorithm. Simulational results demonstrate the superiority of the proposed algorithm in both visual fidelity and numerical measures.

This paper is organized as follows: section 2 gives a brief introduction of SR problem. A novel learning dictionary algorithm (R-KSVD) is presented in section 3. Section 4 discusses how to estimate better HR residual image in the Back Projection. The experimental results are shown in Section 5. Finally, the conclusions are provided in Section 6.

2 Super-Resolution from One Single Image

Super-resolution remains extremely ill-posed because of many high-resolution images X satisfying the above reconstruction constraint. Only under reasonable assumptions and prior knowledge conditions, the SR problem is an inverse problem. We regularize the problem via the sparse representation prior on small patches x of X. The patches x of the HR image X can be represented as a sparse linear combination in a HR sparse dictionary $D_h \in R^{N\times K}$, namely:

$$x \approx D_h \alpha \text{ for some } \alpha \in R^K \text{ with } \|\alpha\|_0 \ll L \tag{1}$$

where K represents the number of atoms in the dictionary, L is the sparsity constraint parameter.

Next, we will recover the HR image patch from the LR image patch. Here, the coupled dictionaries are learned by R-KSVD. We generate the image patch pairs $P = \{x_h, y_\ell\}_i$ from one single LR image as like Glasner [6], here applies guided filter, the first-order and second-order derivatives to extract features $\{y_\ell\}_i$, and $\{x_h\}_i$ are the corresponding HR image patches. We combine these two objectives together to keep the consistent of two dictionaries, and then learn the coupled dictionaries by forcing the LR and HR sparse representations to operate the same codes, which can be formulated as follows:

$$\min_{\{D_\ell, D_h, \alpha\}} \frac{1}{N} \left\| y^\ell - D_\ell \alpha \right\|_2^2 + \frac{1}{M} \left\| x^h - D_h \alpha \right\|_2^2 + \lambda \left(\frac{1}{N} + \frac{1}{M} \right) \|\alpha\|_1 \tag{2}$$

where, M and N are the dimensions of the low-resolution image features and high-resolution image patches in vector form. Equation (2) can be rewritten as

$$\min_{\{D_\ell, D_h, \alpha\}} \left\| \hat{Z} - \hat{D}\alpha \right\|_2^2 + \lambda \left(\frac{1}{N} + \frac{1}{M} \right) \|\alpha\|_1 \tag{3}$$

where

$$\hat{Z} = \begin{bmatrix} \frac{1}{\sqrt{N}} y^\ell \\ \frac{1}{\sqrt{M}} x^h \end{bmatrix}, \qquad \hat{D} = \begin{bmatrix} \frac{1}{\sqrt{N}} D^\ell \\ \frac{1}{\sqrt{M}} D^h \end{bmatrix} \tag{4}$$

We can reconstruct the HR image by $X \approx D_h \alpha$.

3 R-KSVD Algorithm

Recently, numerous algorithms of dictionary learning mainly focus on training an over-complete dictionary in a single feature space. Many methods follow an iterative scheme that alternates between updates of α and D . In the first phase it optimizes for α while keeping D fixed, and in the second phase D is updated using the computed coefficient matrix α . The iteration repeats until some stopping criterion is satisfied. A wide range of dictionary learning algorithms have been developed in the field of SR [11-12]. K-SVD algorithm is very popular for its high efficiency in dictionary learning, which updates the coefficient matrix α by OMP in the first phase [10].

The main idea of the OMP algorithm is choosing the column of D by greedy iterative method, which makes the chosen column and the present redundant vector related to the greatest extent, and then subtracts the related part from measurement vector. The OMP repeats the procedure above until convergence condition meets. In terms of the greatest extent, the OMP algorithm may not obtain the optimal solution.

We propose a novel approach of updating coefficient matrix named R-OMP, which relaxes constrain of choosing the column of D . At first, we define the relaxation factor a . Our method randomly chooses one column of D from the candidate columns, which their values of the greatest extent are beyond a in training process. That is to say, we find slightly smaller values than the maximum value of the inner product of residual e and the columns of sensing matrix d_j , and then randomly choose one. This paper proposes a novel dictionary learning algorithm, which apply R-OMP to update coefficient matrix α while keeping D fixed. The algorithm called

Relaxation K-SVD (R-KSVD). The complete algorithm is described in Table 1. When we use K-SVD algorithm to obtain the dictionary, there are two schemes to control the precision of the algorithm. One is to constrain the representation error, and the other is to constrain the number of nonzero entries in the sparse representation coefficients. In R-KSVD method, we use the latter scheme because it is required in the R-OMP.

Table 1. The R-KSVD algorithm

Input: sampling vector y , the sparsity constraint parameter L , the relaxation factor a ;

Output: the sparse matrix α^* , effective dictionary D ;

Initialization: initial dictionary D_0 , the residual $e_0 = y$, index set $\Delta_0 = \phi$, $t = 1$;

R-KSVD algorithm is mainly two steps:

a. The sparse approximation step: given D , we estimate α , using a sparse approximation algorithm by R-OMP;

R-OMP: execute steps 1 to 5 until convergence:

Step 1: find slightly smaller values (the value beyond a magnification of the maximum value)than the maximum value of the inner product of residual e and the columns of sensing matrix d_j , and then randomly choose one column from the candidates, the corresponding foot mark is θ ;

Step 2: renew the index set $\Delta_t = \Delta_{t-1} \cup \{\theta\}$, the sensing matrix

$$D_t = [D_{t-1}, d_j] ;$$

Step 3: solve $\alpha_t^* = \min \left\| y - D_t \alpha * \right\|_2$ by least-square method;

Step 4: renew the residual

$$e_t = y - D_t \alpha_t^*, t = t + 1;$$

Step 5: if $t > L$, stop the iteration, else do step 1.

b. The dictionary update step: use SVD to jointly re-estimate each atom and its nonzero coefficients to minimize the cost function (3).

4 Back Projection Algorithm

In this section, the traditional Iterative Back Projection (IBP) approach is presented. This method is the backbone of the proposed algorithm. Though simple IBP method can minimize the restoration error significantly in iterative manner and give reasonable performance, it projected the error back without any edge guidance. In this paper, we apply guided filter to estimating better HR residual image in the Back Projection.

Our IBP procedure starts with the input LR image Y .The initial HR image $X(0)$ can be recovered from the input LR image by the proposed SR algorithm. The simulated LR image $Y^{(n)}$ is evaluated by the SR result, as shown in equation (5).

The final HR image is estimated by guided filter for edge preserving and back projecting the error between simulated LR image and the input LR image. The iteration repeats until some stopping criterion is achieved.

The estimated HR image after n iterations is given by:

a) The recovered X should be consistent with the input image Y . The simulated LR image $Y^{(n)}$ can be viewed as a blurred and down sampled version of the HR $X^{(n)}$:

$$Y^{(n)} = DHX^{(n)} \tag{5}$$

where H represents a blurring filter, and D the down sampling operator.

b) Computing the error from LR images as

$$E^{(n)} = (Y - Y^{(n)})\,U^S \tag{6}$$

where, U^S is up-sampling by factor S , Y is initial input LR image, $Y^{(n)}$ is simulated LR image of the nth iteration, $E^{(n)}$ is error estimation.

c) Updating the HR image

$$X^{(n)} = X^{(n-1)} + E^{(n)} \tag{7}$$

$$X^{(n)} = guidedfilter(X^{(n)}, X^{(n)}, r, \gamma) \tag{8}$$

where, r is a radius of a square window, γ is the regularization parameter, $X^{(n)}$ is HR image of the nth iteration. The iteration repeats until some stopping criterion is achieved. The illustration of the complete method is described in Fig 1.

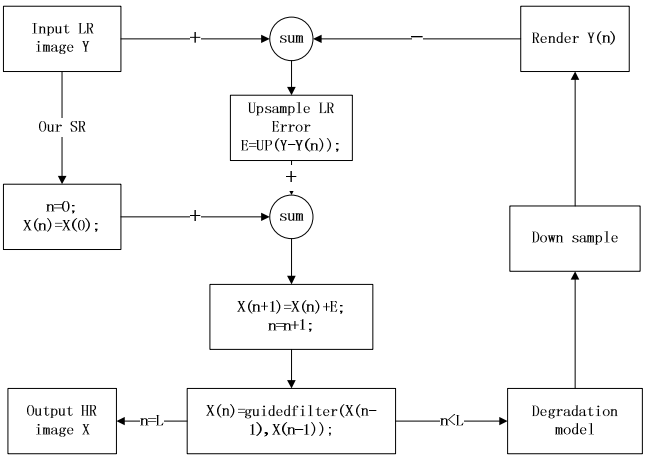


Fig. 1. The process of back projection: firstly, we up-sample the input image by the proposed SR algorithm as the original HR image; then we calculate the error between the input image and the simulated LR image, and add it to the HR image; finally, we obtain the further HR image by guided filter. It repeats the procedure above until iterations meets.

5 Experimental Results

In simulations, we magnify the low-resolution test image by a factor 4 in image super-resolution experiments. By experimental results, it is fit to define the relaxation factor as $a = 0.9$. We define the square window of a radius $r = 2$ and regularization parameter $\gamma = 0.01$ as commonly used in the literature [8], and set sparse parameter $\lambda = 0.1$ as Yang. [15]. Firstly, we extract the features of low-resolution image by guided filter, the first-order and second-order derivatives. Then the image database from different scales and directions is generated [6]. We use 2×2 low-resolution image patches with overlap of one pixel between adjacent image patches in the low-resolution images. Totally 1024 dictionary atoms are learned with each atom of size 2×2.

We apply our method to the luminance channel for color images, because humans are more sensitive to luminance space in visual sense. We interpolate the color layers (Cb, Cr) using bicubic (Bic) interpolation. Numerous natural low-resolution images are simulated, but we only show five images in Fig 2.We apply different algorithms including nearest neighbor (NN) interpolation, bicubic interpolation, Glasner algorithm [6] and our proposed algorithm to reconstruct the HR image. The experimental results are compared from two aspects, RMSE and visual effect. From the Table 2 and Fig 3, we can conclude that our algorithm is superior to the other algorithms in both visual fidelity and numerical measures. What's more the proposed SR method reduces the run time to some extent.

Fig. 2. Five low-resolution test images. From left to right: building; raccoon; girl; visual chart; flower

Table 2. Comparison results of the RMSE, and Run time cost of the different algorithms

RMSE	building	raccoon	girl	visual chart	flower
NN	23.1935	10.3028	6.4315	38.5656	4.2923
Bic	19.106	8.9977	5.2137	29.4436	3.1928
Glasner [6]	18.2757	8.8952	4.9755	19.4362	2.9632
Our algorithm	18.1713	8.3894	4.8242	19.3914	2.9015
Time cost(s)					
Glasner [6]	776.73	2634.59	1875.23	779.43	1472.96
Our algorithm	763.66	2134.69	1400.26	755.05	1302.54

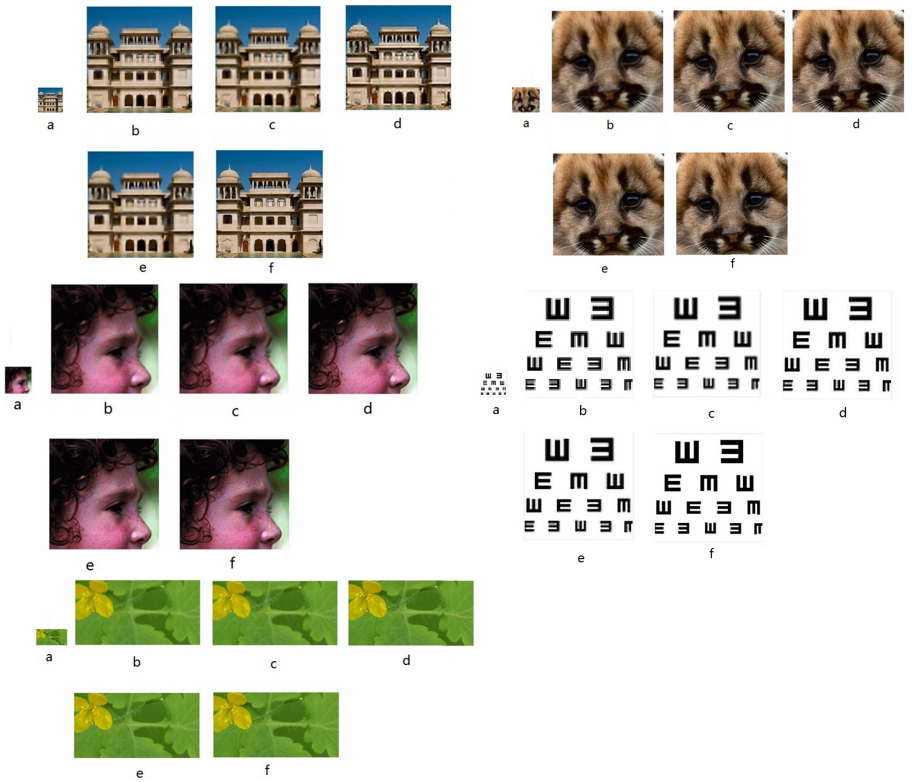


Fig. 3. SR results of building, raccoon, girl, visual chart, flower. (a) a LR image; (b) NN result; (c)Bic result;(d)Glasner result;(e)our result;(f)original image.

6 Conclusions

In this paper, we propose a new method for image super-resolution from only one LR input image. Our algorithm reduces the reconstruction error significantly and avoids the influence of extra database while maintaining comparable image quality. Experimental results demonstrate the superiority of the presented method. In the future, we will consider how to extract more effective features and how to reduce the computation time in further.

Acknowledgments. The authors confirm that the research was supported by National Natural Science Foundation (No.61170109, No.61100119), and Science and Technology Planning Project of Zhejiang Province (No.2012C21021, LY13F020015), China.

References

1. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
2. Goto, T., Suzuki, S., Hirano, S.: Fast and High Quality learning Based Super-Resolution Utilizing TV Regularization Method. In: IEEE International Conference on Image Processing, pp. 1185–1188 (2011)
3. Tang, Y., Pan, X.L., Yuan, Y., Yan, P.K., Li, L.Q., Li, X.L.: Local Semi-Supervised Regression for Single-Image Super-Resolution. In: Multimedia Signal Processing (MMSP), pp. 1–5 (2011)
4. Yang, S., Wang, M., Chen, Y., Sun, Y.: Single-Image Super-Resolution Reconstruction via Learned Geometric Dictionaries and Clustered Sparse Coding. IEEE Transactions on Image Processing, 4016–4028 (2012)
5. Jiang, J.J., Hu, R., Han, Z., Huang, K., Lu, T.: Efficient Single Image Super-Resolution Via Graph Embedding. In: IEEE International Conference on Multimedia and Expo, pp. 610–615 (2012)
6. Glasner, D., Bagon, S., Irani, M.: Super-Resolution from a Single Image. In: IEEE 12th International Conference on Computer Vision, pp. 349–356 (2009)
7. Chen, S.F., Gong, H.J., Li, C.H.: Super resolution from a single image based on self-similarity. In: Computational and Information Sciences (ICCIS), pp. 91–94 (2011)
8. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
9. Tibshirani, R.: Regression shrinkge and selection via the Lasso. J. Royal Statist., 267–288 (1996)
10. Rubinstein, R., Peleg, T., Elad, M.: Analysis K-SVD. IEEE Transactions on Signal Processing & Analysis, 1–17 (2012)
11. Zhou, F., Yang, W.M., Liao, Q.M.: Single image super-resolution using incoherent sub-dictionaries learning. IEEE Transactions on Consumer Electronics, 891–897 (2012)
12. Yang, J., Wang, Z., Cohen, Z.S., Huang, T.: Coupled Dictionary Training for Image Super-Resolution. IEEE Transactions on Image Processing, 3467–3478 (2012)
13. Kim, K., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1127–1133 (2010)
14. Tang, Y., Yan, P., Yuan, Y., Li, X.: Single-image super-resolution via local learning. International Journal of Machine Learning and Cybernetics, 15–23 (2011)
15. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process, 2861–2873 (2010)

Bilateral Multi-issue Parallel Negotiation Model Based on Reinforcement Learning

Lihong Chen, Hongbin Dong*, Qilong Han, and Guangzhe Cui

Department of Computer Science and Technology,
Harbin Engineering University, Harbin, China
donghongbin@hrbeu.edu.cn

Abstract. This paper proposes a bilateral multi-issue parallel negotiation model based on reinforcement learning. Considering the equality of both sides and that both negotiators refuse to give more information for their own interests, it introduces a mediator agent as the mediation mechanism. It considers the correlation between quantity and price simultaneously, and uses reinforcement learning to generate the optimal behavioral strategy. Comparing with 'A simultaneous multi-issue negotiation through autonomous agents', the experimental results show that the proposed method has decreased little in the negotiation joint utility, but has decreased significantly in the negotiation times , and has also improved the equality of both negotiators.

Keywords: Automated negotiation, Reinforcement learning, Mediator agent, Related issues.

1 Introduction

Negotiation is an important method to realize online e-commerce and also an important goal of designing software agent.

Today people mainly research on multi-agent negotiation under the incomplete information, and that agents could dynamically change their negotiating tactics through learning environment knowledge. Reinforcement Learning[5] is an efficient machine learning method, learning from environment state to action mapping to maximize the cumulative reward obtained from environment , finding the optimal behavioral strategy by trial and error. Without learning the markov decision model (T function and R function), Q-learning is one of the most commonly used algorithm in the reinforcement learning.

Tianhao Sun et al., proposed a reinforcement learning negotiation strategy based on bayesian Classification[1].Xin Sui et al., proposed an multi-agent negotiation strategy based on Q-Learning[2], Linlan Zhang et al., put forward a multi-round sealed-bid negotiation mechanism based on mediator agent[3]. According to the advantages and disadvantages of above three negotiation models, this paper proposes a bilateral

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 40–48, 2013.

multi-issue parallel negotiation model based on reinforcement learning which uses Q-learning algorithm to generate the optimal proposals.

2 Negotiation Framework

This paper is mainly negotiating on the price and quantity this two issues, and introduces an intermediary agent as mediation mechanism, using utility function to evaluate the proposals, thus this part briefly describes the negotiation framework from the following two aspects.

2.1 Formal Definition of the Model

This section defines some basic symbols to represent the problem , using the general negotiation model of Agent, described as : Neg=<G,A,D,U,T>.

Where G is the set of negotiating agents, including the buyer agent b, the seller agent s and the mediator agent m. A represents the set of negotiation issues, including the issue price p and the isssue quantity q. D is the value interval of negotiation issues, [IPb, RPb] and [RPs, IPs] respectively represent the price interval of the buyer agent and the seller agent. U is the utility of negotiation agents. $U = \{U_b, U_s, UR_b, UR_s\}$, respectively represent the utility of the buyer and the seller , the reservation utility of the buyer and the seller . T represents the deadline of negotiation, T={Tb,Ts}, respectively represent the deadline of the buyer and the seller.

There are some symbols not listed, in order to understand easily we will introduce the corresponding symbols when give the related definitions.

2.2 Utility Evaluation

This article only negotiates on the two issues p and q, and the two issues are interlinked, expected utility is a dual function of these two issues. We assume each agent incurs a fixed cost, c. Defined as in the reference [3], the buyer and the seller's expected utility are as follows:

(1) Seller agent

$$U_s(p,q) = (p - RPs).q - c \tag{1}$$

(2) Buyer agent

The buyer determines its purchase quantity q according to the market demand x. We assume x is a uniform probabilistic distribution on [a, b]. Let p_0 denote the sale price.If q > x, the buyer will sell the products with a low price αp_0. Assuming that αp_0 is lower than purchase price p, and p_0 is higher than p. Since p ∈ [IPb, RPb], accordingly, αp_0 <IPb and p_0 >RPb. Then for a given proposal < p, q >, the buyer agent's utility function f is defined as follows:

$$f(p,q,x)=\begin{cases}(p_0-p).q & if \quad q\le x\le b,\\ (p_0-p).x+(\alpha p_0-p).(q-x) & if \quad a\le x\le q.\end{cases}$$

Note that x is a variable, then the buyer's expected utility is defined as follows:

$$\begin{aligned}U_b(p,q)&=\int_a^b f(p,q,x).\frac{1}{b-a}dx-c=\frac{1}{b-a}\\ &(\int_a^q (p_0-p).x+(\alpha p_0-p).(q-x)dx+\int_q^b (p_0-p).qdx)-c=\frac{1}{b-a}\\ &.\left(-\frac{(1-\alpha)p_0}{2}.q^2+(p_0(b-\alpha a)-p(b-a))q-\frac{(1-\alpha)p_0}{2}.a^2\right)-c.\end{aligned} \tag{2}$$

3 Negotiation Strategy Based on Reinforcement Learning

This paper uses Q-learning algorithm to generate the optimal proposed strategy.In the process of generating proposals, we need consider the time belief, issue belief(i.e.probability distributions knowledge), etc. Therefore, following the reference [1] here gives three definitions:

Definition 1. Time belief refers to the probability that negotiation agent thinks the opponent accept its offer, which is correlated with time. Which bfs(t), sfb(t) respectively represent the time belief of the buyer and the seller, which generally falls into three categories: increasing function (e.g., bfs = t/Tb), decreasing function (e.g., bfs = 1 - t/Tb), and constant function (e.g., bfs = 0.5). Here we have bfs = 1 - t/Tb and sfb=1-t/Ts.

Definition 2. Price belief refers to the cognition of probability distribution that negotiation agent thinks the issue p in its value interval . Which bps(t), spb(t) respectively represent the price belief of the buyer and the seller, which bps(t)=1/(p_0 -IPb), spb(t)=1/(IPs-RPs).

Definition 3. Quantity belief refers to the cognition of probability distribution that negotiation agent thinks the issue q in its value interval .

bqs(t), sqb(t) respectively represent the quantity belief of the buyer and the seller. Assuming that the value interval of q of both agents is [a,b], then bqs(t)=sqb(t)=1/(b-a).

The both agents submit their offers to the mediator agent simultaneously. Only after the negotiation achieves success, can it obtain the corresponding reward value r. Expected utility is the reward values acquired after success. Assuming when the negotiation achieves success, the agreement price is p^T , and the agreement quantity is q^T , then the reward values of both agents are as follows:

$$\text{Seller:}\quad r_s=U_s(p^T,q^T)=(p^T-RPs).q^T-c \tag{3}$$

$$\text{Buyer:}\quad r_b=U_b(p^T,q^T) \tag{4}$$

The Q function based on Q-learning is defined as:

$$Q(s(t), p(t), q(t)) = r(s(t), p(t), q(t)) + \gamma \max_{p(t+1),q(t+1)} Q(\delta(s(t), p(t), q(t)), p(t+1), q(t+1)) \tag{5}$$

Where, γ is the time discount rate, and δ() is the state transition function.

Assumes that at round t the negotiation succeeds, and obtains the reward value Qe. Here for convenience, the t-th time proposal is called the t-th stage proposal. Then, according to the definition of Q function , suppose the negotiation succeeds at the first proposal , so the Q function value at the first stage is Q(s(1),p(1),q(1))=Qe; The negotiation succeeds at the second proposal, Q(s(1),p(1),q(1))=γQe; ..... ;The negotiation succeeds at the t-th time proposal, Q(s(1),p(1),q(1))= r^{t-1} Qe. So the seller's average reward value at the first stage is as follows:

$$\overline{Qse}(s(1), p(1), q(1)) = \frac{\sum_{i=1}^{Ts} sfb(i)\gamma^{i-1} Qse}{Ts} \tag{6}$$

The seller's average reward value at stage t can be deduced as follows:

$$\overline{Qse}(s(t), p(t), q(t)) = \frac{\sum_{i=t}^{Ts} sfb(i)\gamma^{i-t} Qse}{Ts - t + 1} \tag{7}$$

Similarly, the buyer's average reward value at stage t is:

$$\overline{Qbe}(s(t), p(t), q(t)) = \frac{\sum_{i=t}^{Tb} bfs(i)\gamma^{i-t} Qbe}{Tb - t + 1} \tag{8}$$

Suppose the negotiation achieves success at round T, the expected rewards of both agents are as follows:

The reward of seller agent：

$$Qse = \iint_D r_s.spb.sqb.dp^T dq^T = \int_a^b \int_{RPs}^{IPs} r_s . \frac{1}{IPs - RPs} . \frac{1}{b - a} . dp^T dq^T \tag{9}$$

Similarly, the reward of buyer agent：

$$Qbe = \iint_D r_b.bps.bqs.d\,p^T dq^T = \int_a^b \int_{IPb}^{p_0} r_b . \frac{1}{p_0 - IPb} . \frac{1}{b - a} . dp^T dq^T \tag{10}$$

In summary, the bidding strategy of seller agent is:

$$p_s(t) = \frac{\overline{Qse} + c}{q_s(t)} + RPs \tag{11}$$

And the bidding strategy of buyer agent is:

$$p_b(t) = \frac{b-\alpha a}{b-a}.p_0 - \frac{(1-\alpha)p_0((q_b(t))^2 + a^2)}{2(b-a)q_b(t)} - \frac{\overline{Qbe}+c}{q_b(t)} \tag{12}$$

To obtain the value of p, here draws out five propositions in reference [3].

Proposition 1. At the initial stage of the negotiation, the optimal proposal $< p_s(0), q_s(0) >$ of seller agent satisfies the following equations: $p_s(0) = IPs, q_s(0) = b$. (13)

Proposition 2. At the initial stage of the negotiation, the optimal proposal $< p_b(0), q_b(0) >$ of buyer agent satisfies the following equations: $p_b(0) = IPb, q_b(0) = Ceiling(((b-\alpha a)p_0 - (b-a)p_b(0))/((1-\alpha)p_0))$. (14)

Proposition 3. At negotiation round t($t \geq 1$), the optimal proposal $< p_s(t), q_s(t) >$ of seller agent satisfies the following equations: $p_s(t) = \frac{\overline{Qse}+c}{q_s(t)} + RPs$, $q_s(t) = b$. (15)

Proposition 4. At negotiation round t($t \geq 1$), the optimal proposal $< p_b(t), q_b(t) >$ of buyer agent satisfies the following equations:

$$p_b(t) = \frac{b-\alpha a}{b-a}.p_0 - \frac{(1-\alpha)p_0((q_b(t))^2 + a^2)}{2(b-a)q_b(t)} - \frac{\overline{Qbe}+c}{q_b(t)},$$

$$q_b(t) = Ceiling\left(\sqrt{a^2 + 2(b-a)(\overline{Qbe}+c)/((1-\alpha)p_0)}\right). \tag{16}$$

Proposition 5. When the quantity $q = Ceiling(((b-\alpha a).p_0 - (b-a).RPs)/((1-\alpha)p_0))$ the joint utility $U = U_s + U_b$ is the maximal.

The detailed proof of the above propositions can be seen in reference [3].

4 Negotiation Algorithm

This part describes the negotiation algorithm of the negotiation model shown as below:

(1) Before the negotiation starts, both agents submit their value scopes of quantity. The mediator agent determines the value interval of quantity [a, b] through market analysis and negotiation with both agents, and informs both agents.

(2) At the initial stage of the negotiation, the buyer agent and seller agent submit their initial offers to the mediator agent simultaneously. The seller's initial proposal is given according to proposition 1, and the buyer's initial proposal is given according to proposition 2.

(3) At each round of the negotiation , the seller agent and the buyer agent respectively generate their optimal proposals according to proposition 3 and proposition 4.

(4) The mediator agent judges whether there is a trading opportunity when receives the offers from both agents. According to the propositions before (proposition 1-4), there always exists $q_s > q_b$ .So unless $p_b > p_s$, otherwise there is no trading opportunity.

Suppose at round T, there exists $p_b(T) > p_s(T)$. The mediator agent determines the trading opportunity and informs both agents to submit all of the proposals on the interval $[q_b(T), q_s(T)]$ at round T+1 . All the proposals generated must embody the equal utility. If there does not exist $p_b(T+1) > p_s(T+1)$ of all the proposals under the same q, the mediator agent will inform both agents into the next round. Otherwise the proposal with the maximum difference on $p_b(T+1) - p_s(T+1)$ at the same q determines initial agreed proposal. And the initial agreed price is $(p_b(T+1) + p_s(T+1))/2$.

If there is no trading opportunity, the mediator agent will inform both agents into the next round.

(5) The buyer and the seller judge whether to accept the proposal according to their respective reservation utility when receive the initial agreement proposal from the mediator agent . If the initial agreement proposal is greater than the reservation utility then accepts the proposal, or rejects the proposal and the mediator agent will inform both agents into the next round. If both agents accept the proposal, an agreement is reached , and it is the final agreement proposal.

(6) The negotiation process continues until an agreement is reached or one of the negotiators terminates the process for reaching deadline.

5 Experimental Design and Analysis

In order to verify the effectiveness and the efficiency of the method, this part has carried out an experiment on the platform of VC++ , and carried on a comparative analysis with the experimental results of the literature [3]. Firstly ,we initialize some parameters. For the sake of convenience of the contrast test , we follow the parameter setting in the reference [3], as shown in table 1.

Table 1. Parameters Setting

Parameters	Buyer	Seller
RP		[83 75 72 60]
IP	[68 62 60 55]	[110 100 97 86]
UR	[605.5 655.5 665.5 712.5]	[270 240 230 200]
Td	12	10
α	0.5	
p_0	100	
a	100	
b	150	

We have carried out a comparative analysis on four groups of datas. As shown in table 1, both agents have four groups of datas, accordingly marked as N1, N2, N3 and N4 in turn. It can be seen that the column of the reservation price of the buyer agent is tagged empty, this is because the utility and the evaluation of an offer of the buyer agent both have nothing to do with the reservation price. so we needn't initialize it.

Figure 1 shows the negotiation process of N4 on the two negotiation models. Because the price p is the only variable judging whether there is a trading opportunity, so here we only show the negotiation process of the price varying with the negotiation rounds(t). Where s1, b1 respectively represent the seller agent and the buyer agent of the model in the paper [3], and s2, b2 respectively represent the seller agent and the buyer agent of the model this paper.It can be seen that the model in paper [3]: when t = 6, the buyer's price(76.451) is greater than the seller (73.663) , for the first time there is an trading opportunity. When t = 7 the mediator agent calculates the initial negotiation agreement < 75.169, 139 >. Both agents judge out that the initial agreed proposal is greater than their respective reservation utility, then accept the proposal. An agreement is reached , and the final agreed proposal is < 75.169, 139 >. And the model of this paper: when t = 1, the buyer's price (87.6264) is greater than the seller's price (66.3531), for the first time there is an trading opportunity. And reaches an agreement at t = 2 , the agreed result is <77.5982, 121 >. It can be seen that the negotiation model of this paper on negotiation N4 is significantly less than the model in paper [3] in the negotiation rounds.

Here conducts a performance analysis from two aspects to evaluate the negotiation result on the negotiation N4 :

(1) Joint utility: According to the proposition 5 it can be calculated that the optimal joint utility is 4798 where q = 140, and the joint utility of the negotiation result is $U_b + U_s$ = 2489.12 + 2128.38 = 4617.5, which obtains the optimal percentage of 4617.5/4798 = 96.28%, which is almost optimal. The result is also not much less than the result (99.99%) in literature [3].

(2) Equality: The utility of both agents this paper is normalized as follows:

$u_b = U_b / (U_b + U_s) = 2489.12/4617.5 = 0.53906,$

$u_s = U_s / (U_b + U_s) = 2128.38/4617.5 = 0.46094 .$

Then equality= $|u_b - u_s| = |0.53906 - 0.46094| = 0.07812 < 0.1$. In literature [3], the equality is $|u_b - u_s| = |0.56071 - 0.43929| = 0.12142$. Showing that for the negotiation N4 the method of this paper has also improved in equality.

Table 2. Negotiation Results

Negotiation	Result		Round		Joint utility		Equality	
	QRL	SMN	QRL	SMN	QRL	SMN	QRL	SMN
N1	<91.4007, 117>	fail	3	X	1842.5	X	0.06581	X
N2	<86.1459, 118>	<82.1817, 126>	3	10	2786	2810	0.05657	0.35660
N3	<84.2917, 119>	<80.2558, 126>	3	10	3149.5	3188	0.07178	0.34803
N4	<77.5982, 121>	<75.1690, 139>	3	8	4617.5	4797.5	0.07812	0.12142

Table 2 shows the negotiation results of the four groups of negotiation on the two negotiation models. Which QRL represents the model of this paper, SMN represents the model in paper[3].The fourth column shows that this two models have little difference in joint utility. The third column shows that the negotiation model of this paper is significantly less than the model SMN in the negotiation rounds. And the fifth column shows that our modelhas also improved in equality than the model SMN.

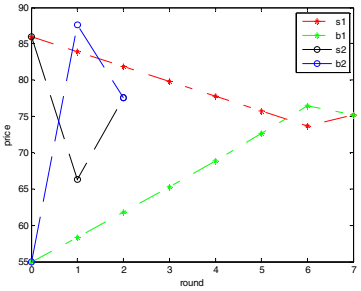


Fig. 1. Negotiation process of N4

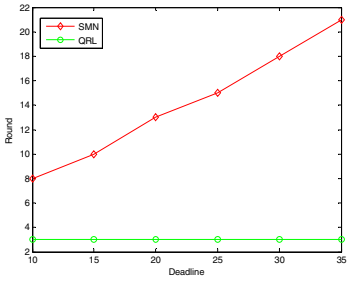


Fig. 2. Rounds of negotiation

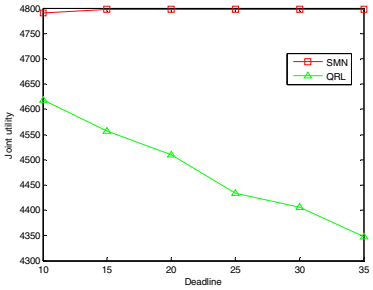


Fig. 3. Joint utility of negotiation

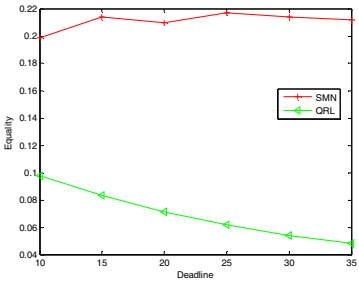


Fig. 4. Equality of negotiation

Figure 2, figure 3, figure 4, respectively shows the changing curves that the negotiation rounds, the joint utility of negotiation , and the equality of negotiation vary with the change of the deadline under the two negotiation models. Here sets that the deadlines ({10,15,20,25,30,35})of the buyer and the seller are the same. Figure 2 demonstrates that the negotiation rounds of SMN increases with the increasing deadline, while the negotiation rounds of QRL has no change and been being 3. Which shows the higher stability of the proposed method on the negotiation rounds, and it can be seen that the negotiation rounds of QRL is significantly less than the SMN. So our method obviously improves the negotiation rate. And in figure 3, the joint utility of SMN has almost no change with the increasing deadline, but of QRL decreases along with the increasing deadline.Which demonstrates that the stability of the proposed method on the joint utility is not high, and the performence in joint utility is also slightly inferior to the SMN. In Figure 4, the equality of SMN has little change with the change of the deadline , while the equality of QRL has the trend to decrease and is also better than that of SMN.

6 Conclusion

This paper proposes a bilateral multi-issue parallel negotiation model based on reinforcement learning, introducing a mediator agent as the mediation mechanism. It uses Q-learning algorithm to generate the optimal behavioral strategy.Comparing with the paper[3], the experimental results show that the proposed method has decreased little in the negotiation joint utility, but decreased significantly in the negotiation times , and has also improved the equality of both negotiators.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China under Grant(60973075,61272186), and supported by Fundamental Research Funds for the Central Universities(HEUCF100607).

References

1. Sun, T., Chen, F., Zhu, Q., Cao, F.: Reinforcement learning negotiation strategy based on bayesian classification. Chinese Journal of Computer Science, 227–229 (2011)
2. Sui, X., Cai, G., Shi, L.: Multi-agent negotiation strategy and algorithm based on Q-Learning. Chinese Journal of computer engineering, 198–200 (2010)
3. Zhang, L., Song, H., Chen, X., Hong, L.: A simultaneous multi-issue negotiation through autonomous agents. European Journal of Operational Research, 95–105 (2010)
4. Esmaeili, A., Mozayani, N.: Improving multi-agent negotiations using multi-objective PSO algorithm. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010, Part I. LNCS, vol. 6070, pp. 92–101. Springer, Heidelberg (2010)
5. Gao, Y., Chen, S., Lu, X.: A research overview on reinforcement learning. Chinese Acta Automatica Sinica, 86–100 (2004)
6. Chohra, A., Bahrammirzaee, A., Madani, K.: Time and Personality Dependent Behaviors for Agent Negotiation with Incomplete Information. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 352–362. Springer, Heidelberg (2011)
7. Hadidi, N., Dimopoulos, Y., Moraitis, P.: Argumentative alternating offers. In: McBurney, P., Rahwan, I., Parsons, S. (eds.) ArgMAS 2010. LNCS, vol. 6614, pp. 105–122. Springer, Heidelberg (2011)
8. Fang, F., Wong, T.N.: Applying hybrid case-based reasoning in agent-based negotiations for supply chain management. Expert Systems with Applications, 8322–8332 (2010)

Learning to Detect the Subway Station Arrival for Mobile Users

Kuifei Yu[1], Hengshu Zhu[2], Huanhuan Cao[3],
Baoxian Zhang[1], Enhong Chen[2], Jilei Tian[3], and Jinghai Rao[3]

[1] University of Chinese Academy of Sciences, Beijing, China
yukuifei@gmail.com, bxzhang@ucas.ac.cn
[2] University of Science and Technology of China, Hefei, China
zhs@mail.ustc.edu.cn, cheneh@ustc.edu.cn
[3] Nokia Research Center, Beijing, China
{happia.cao,jilei.tian,jinghai.rao}@nokia.com

Abstract. The use of traditional positioning technologies relies on the underlying infrastructures. However, for the subway environment, such positioning systems may not be available for the positioning tasks, such as the detection of the train arrivals for the passengers in the train. An alternative way is to exploit the contextual information available in the mobile devices of subway riders. To this end, in this paper, we propose to exploit multiple contextual features extracted from the mobile devices of subway riders for precisely detecting train arrivals. Along this line, we first investigate potential contextual features which may be effective to detect train arrivals according to the observations from sensors. Furthermore, we propose to explore the maximum entropy model for training a train arrival detector by learning the correlations between the contextual features and the events of train arrivals. Finally, we perform extensive experiments on several real-world data sets. Experimental results clearly validate both the effectiveness and efficiency of the proposed approach.

Keywords: Subway Arrival Detection, Mobile Users, Smart Cities.

1 Introduction

Advances in smart mobile technologies have enabled unprecedented capabilities for context sensing. In this paper, we study the problem of detecting subway arrivals for the passengers in the train by exploiting the contextual information collected from the sensors in mobile devices. This problem is also an important context recognition problem which has a wide range of potential applications. In the following, we provide a case to intuitively illustrate the application of detecting train arrivals for improving the user experiences for subway riders.

Example 1 (Subway Arrival Reminding Service.). Jack enjoys reading, catnapping, or building castles in the air on the subway train so much that he involuntarily ignores recurring subway broadcasts, and he had missed his destination station from time to time. However, with a subway reminding application on his

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 49–57, 2013.

mobile phone enabled by the subway arrival detection technology, Jack can be reminded before the arrival of his destination station.

However, it is a nontrivial task to detect the subway arrivals, because most traditional accurate positioning technologies, such as GPS positioning, cell tower triangulation, and Wifi local positioning, are not available in the subway environment. To be specific, first, it is well known that mobile devices cannot receive GPS signals in the subway system. Second, cell tower triangulation positioning [2, 7] cannot directly work for detecting subway arrivals either, because the cell towers serving for subway riders are usually linearly deployed along the subway lines. This is different from the deployment of cell towers on the ground. Actually, even on the ground, the errors of cell tower triangulation positioning systems can be as high as tens of meters. Given the sparse deployment of 4G cell towers in many countries due to the lack of high frequency bands [12], the accuracy of cell tower triangulation positioning will be difficult to be improved in the near future. Third, while it seems that Wifi local positioning [4] is a good alternative approach, there are still many subway stations where there are no Wifi accessing points. Finally, people may argue that the subway operation companies know the accurate position of each train. However, up to now, this real-time information is still difficult to be obtained by the third party applications and services in many countries due to various reasons, such as the security concerns. Therefore, in light of the above discussions, a precise approach for detecting subway train arrivals is crucially needed for the effective development of reminding services in the subway systems.

1.1 Problem Statement

To facilitate the description of the following sections, we first formally define the problem of subway arrival detection as follows.

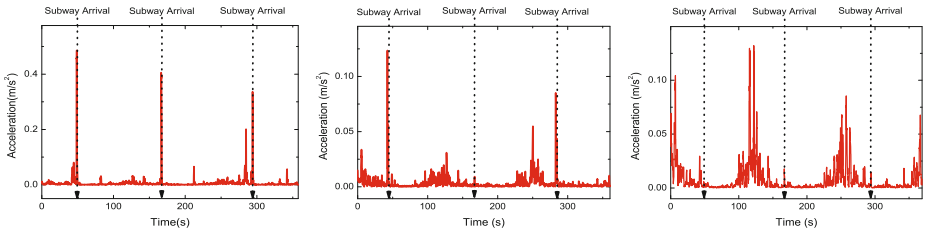
Definition 1 (Station Arrival Detection). Given a detection interval Δt, the objective of subway arrival detection problem is to map the latest contextual feature set $C_t = \{< f^k : v^k >\}$ into two semantic labels $l = \{Arrival, Non\text{-}arrival\}$ every Δt time points, where $< f^k : v^k >$ denotes the *k-th* contextual feature and the corresponding value that is calculated at time point t.

According to the definition, the subway arrival detection problem can be converted to a supervised classification problem. Therefore, this problem is divided into two parts, namely *how to extract effective contextual features from the raw outputs of sensors?* and *how to train an effective station arrival detection model through machine learning technologies?* The solutions for the two sub-problems are presented in the following sections in detail, respectively.

Indeed, there are some related works have been reported in literatures, such as context recognition [3, 8, 9] and accurate positioning technologies [2, 4, 7]. However, to the best of our knowledge, how to detect train arrivals for subway riders is still under-explored. Therefore, in this paper we propose a novel subway arrival detection approach based on the outputs of both 3D accelerometers and GSM sensors to detecting subway arrivals.

2 Contextual Feature Analysis

In this section, we study several contextual features extracted from the outputs of 3D Accelerometers and GSM sensors and preliminarily analyze their effectiveness for detecting subway arrivals.



(a) X-axis (b) Y-axis (c) Z-axis

Fig. 1. The variance of a smart phone's accelerations on (a) X-axis (b) Y-axis and (c) Z-axis when the subway rider passed by three stations

2.1 3D Accelerometer Based Features

The Variance of Single Dimension Accelerations. Figure 1 shows the variances of a smart phone's accelerations on the X-axis v_X, Y-axis v_Y, and Z-axis v_Z when the subway rider passed by three stations. From this figure we can observe that $Variance(v_X)$ and $Variance(v_Y)$ have obvious peaks when the train arrives at stations. By contrast, the peaks of $Variance(v_Z)$ do not seem to be relevant to subway arrivals. Actually, this phenomenon is resulted from the way where the subway rider holds the smart phone.

The Mean Variance of Three Dimension Accelerations. Compared to the variance of single dimension accelerations, the mean variance of three dimension accelerations may be a better contextual feature to indicate subway arrivals since it is not sensitive to the way the subway rider holds the phone. For example, Figure 2 (a) shows the mean value of $variance(v_X)$, $variance(v_Y)$, and $variance(v_Z)$ for a subway rider's smart phone when the train passed by three stations. The figure shows that this feature can indicate subway arrivals well.

2.2 GSM Sensor Based Features

The Shift of Serving Cell-Sites. We define a *Serving Cell-site Shift Time (SCST)* to capture the relevance between subway arrivals and the shift of serving cell-sites, which can be calculated by

$$SCST = t - t_{shift}, \tag{1}$$

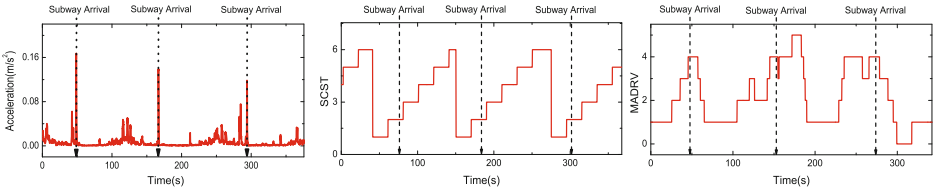


Fig. 2. The change of (a) the mean value of $variance(v_X)$, $variance(v_Y)$ and $variance(v_Z)$, (b) SCST, and (c) MADRV for a subway rider's smart phone when the train passed by three subway stations

where t denotes the time point when a subway arrival detection model makes a detection and t_{shift} denotes the time point of the latest shift of serving cell-sites. Moreover, to guarantee the generality of this feature, we map the value of $SCST$ into five classes according to the observation of real-world data set. Figure 2 (b) shows the changes of SCST for a subway rider's smart phone when the train passed by three subway stations. From this figure we can observe that when the train arrives at a station, the corresponding SCST is always mapped into $Class_2$, which validates the effectiveness of $SCST$.

The Signal Strength of Serving Cell-Site. Intuitively, when a train stays at a station, the corresponding signal strength is stable and relatively high. Therefore, we can directly leverage the RSSI value of signal strength as a contextual feature for detecting subway arrivals. To guarantee the robustness of this feature, we map the RSSI value into 4 levels according to the standard of Android system.

We introduce another feature called *Mean Absolute Deviation of RSSI Value* (MADRV) as follows,

$$MADRV = \frac{\sum_{i=1}^{N}(|V_i - \overline{V}|)}{N}, \tag{2}$$

where V_i indicates the i-th sampled RSSI value before the detection time point t and $\overline{V}$ indicates the average value of the total N sampled RSSI values. The larger MADRV, the more likelihood that the train will arrive at next station soon. Figure 2 (c) shows the change of MADRV for a subway rider's smart phone when the train passed by three subway stations. From this figure we can observe that when the train arrives at a station, the MADRV is always equal to the same value, which validates the effectiveness of the feature.

3 Learning to Detect Subway Arrivals

After contextual feature extraction, the remaining work is to train a detection model **M**, which can integrate multiple effective contextual features for detecting

subway arrivals. Actually, for this problem, given a set of training samples, a lot of supervised classification models can be applied. In this paper, we propose to use the maximum entropy (MaxEnt) classifier for training a detection model.

To be specific, in our problem, given a detection time window Δt and the current time point t, MaxEnt defines the conditional probability of a subway arrival label l (i.e., whether the subway train arrives at a stop or not at t) as

$$P(l|C_t) = \frac{1}{Z(t)} \exp(\sum_i \lambda_i f_i(C_t, l)), \tag{3}$$

where C_t denotes the contextual feature set extracted at the time point t, $f_i(C_t, l)$ denotes a feature function about C_t and l, λ_i indicates the weight of $f_i(C_t, l)$, and $Z(t)$ indicates a normalization factor which equals to

$$Z(t) = \sum_l \exp(\sum_i \lambda_i f_i(C_t, l)). \tag{4}$$

The objective of model training is to learn a set of proper parameters using training data set to maximize the model likelihood. After that we can infer the label l^* according to a contextual feature set $C_{t'}$ as $l^* = \arg\max_l P(l|C_{t'}, \Lambda)$. According to the comparison of algorithms for maximum entropy parameter estimation in [6], we use the most efficient algorithm L-BFGS for model training.

Table 1. The details of collected subway context data

Data set	#3D Acc. Records	#GSM Records	#Subway Arrivals
line 10-1	13,153	15,892	18
line 10-2	6,460	8,752	10
line 10-3	13,097	16,302	18
line 10-4	13,516	16,439	18
line 10-5	7,711	9,984	11
line 5	9,889	11,594	14

3.1 Imbalanced Classification Problem

When we take subway arrival detection as a supervised classification problem, a critical challenge along the line is that the training samples with the label *Arrival* are extremely limited compared with the others. To be specific, in our data sets the average ratio of label *Arrival* to *Non-arrival* is only 0.0063. If we use such imbalanced data to train detection model, the classification accuracy of subway arrival would be very poor. This problem is known as *imbalanced classification problem*, which is well-studied by many researchers [5, 10, 11]. Although MaxEnt is a good model at dealing with imbalanced training data, it still suffers the extreme imbalance of our data sets, as illustrated in our experiments. To solve this problem, in this paper we propose to leverage the two widely used

approaches to imbalance classification, namely, data under-sampling and data over-sampling [5,11]. To the best of our knowledge, how to select the best value of drop rate or duplicate rate is still an open question. Therefore, in this paper we do not give any principles but compare various settings to evaluate the detection accuracy in experiments.

4 Experiments

In this section, we evaluate our approach through extensive experiments on several real-world data sets collected from two major subway lines in Beijing.

4.1 Data Collection and Preprocessing

To study the problem of detecting subway train arrivals through mobile devices, we developed a subway context data collection application for collecting the context data which are potentially useful for detecting subway arrivals, including the outputs of 3D accelerometers and GSM sensors [1].

The application is developed for Google Android 2.3 System. The sampling rate of each sensor is set according to the APIs provided by Google Android system [1]. To prepare the experimental data, we installed the subway context data collection application on a HTC Z710e smart phone and collected many context data from two major subway lines in Beijing, China. To be specific, five data sets were collected from line 10 and one data set was collected from line 5 [13]. The details of the collected data sets are listed in Table 1.

To extract training and test samples from the collected context data, we first determine the detection interval, i.e., Δt, as the interval of 3D accelerometer outputs. Then, for each time point t when a 3D accelerometer output is recorded, we build a sample by extracting the contextual features .

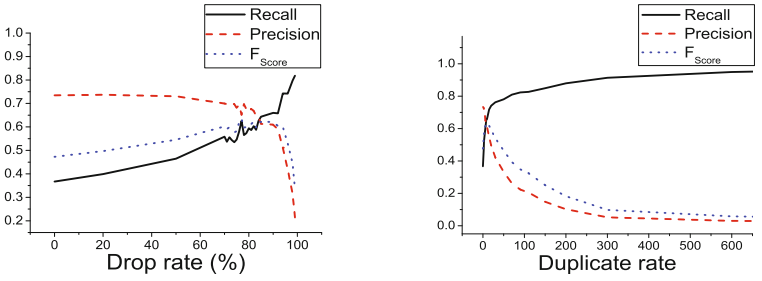


Fig. 3. The detection performance of our approach with respect to different (a) drop and (b) duplicate rates

[1] This data set will be made publicly available soon.

4.2 Benchmark Method and Evaluation Metrics

In this paper we extend a widely used approach in transportation mode recognition [8,9] as the baseline. To be specific, we calculate the mean of $Variance(v_X)$, $Variance(v_Y)$ and $Variance(v_Z)$ for every six 3D accelerometer outputs. If the mean value is larger than a predefined threshold ϕ, we label this detection time as *Arrival*, where ϕ is set to be the minimum mean variance of 3D accelerations for all positive samples in the training data.

Moreover, to study the contribution of 3D accelerometers based features and GSM based features separately, we also evaluate two MaxEnt models with only one kind of features, namely, ME-3D (*MaxEnt with 3D accelerometers based features*) and ME-G (*MaxEnt with GSM sensor based features*), besides the MaxEnt model which combines all contextual features discussed in this paper (denoted as ME-3D-G). All above approaches are implemented by standard C++ and the experiments are conducted on a 3GHZ×4 quad-core CPU, 3G main memory PC.

To evaluate the performance of subway arrival detection, we first use *Recall*, *Precision*, and F_{score} with respect to the *Arrival* label for measuring the outputs of each test approach. We also propose some user experience based metrics with more tolerance for the detections named $Recall_{UX}$, $Precision_{UX}$, and F_{UX}, which are correspondingly defined as $\frac{\#Hit\_Session}{\#Arr\_Session}$, $\frac{\#Hit\_Session}{\#Hit\_Session+\#Error}$, and $2 \times \frac{Recall_{UX} \times Precision_{UX}}{Recall_{UX}+Precision_{UX}}$, where *Arr_Session* denotes a time window which contains the ± 5 seconds of a sample labeled *Arrival* as the ground truth, *Hit_Session* denotes a *Arr_Session* which contains at least a sample labeled *Arrival* by the detection model, and *Error* denotes a sample which is labeled *Arrival* by the detection model but does not fall into any *Arr_Session*.

4.3 The Impact of the Strategy for Reducing Data Imbalance

To solve the problem of imbalanced classification, a standard five-fold cross validation is conducted. Figure 3 (a) shows the detection performance of our approach, i.e., ME-3D-G, with respect to different drop rates in the under-sampling training, where the training samples with the label *Non-arrival* are randomly dropped under each drop rate. From this figure we can observe that *Precision* first slightly drop with the increase of drop rate and sharply drop under a certain drop rate, while *Recall* roughly increase with the increase of drop rate.

Figure 3 (b) shows the detection performance of our approach with respect to different duplicate rates in the over-sampling training. From this figure we can observe that *Precision* consistently drop with the increase of duplicate rate while *Recall* consistently increase until reach an optima with the increase of duplicate rate. Specially, in the following experiments of ME-3D-G, we set the drop rate to be 76% for the under-sampling approach duplicate rate to be 7 for the over-sampling approach.

Similarly, we also study the impacts of different drop rates and duplicate rates to the detection performance of another two MaxEnt Models with different types of features, i.e., ME-3D and ME-G. In the following experiments, we set the drop

rates for ME-3D, ME-G to be 72% and 99%, respectively, and set the duplicate rates for ME-3D, ME-G to be 5 and 55, respectively.

4.4 The Reusability of the Model

Given a subway arrival detection model trained on the context data from a certain subway line, the reusability of the model means its ability of being applied to other context data from the same subway line. In order to study the reusability of our detection models and the baseline method, we leverage a five-fold cross validation to evaluate the detection performance of each approach for five data sets collected from line 10.

Table 2. The average detection performance of each approach in the five-fold cross validation for the data sets collected from line 10

		Recall	*Precision*	F_{score}	$Recall_{UX}$	$Precision_{UX}$	F_{UX}
ME-G	Under-Sampling	0.5967	0.0136	0.0265	0.9647	0.0281	0.0545
	Over-Sampling	0.2394	0.0118	0.0225	0.6354	0.0425	0.0784
ME-3D	Under-Sampling	0.4776	0.2652	0.3370	0.7699	0.3439	0.4699
	Over-Sampling	0.5286	0.2452	0.3339	0.8379	0.3114	0.4529
ME-3D-G	Under-Sampling	0.5793	**0.6926**	0.6182	0.9085	**0.8611**	**0.8740**
	Over-Sampling	0.6262	0.6425	**0.6224**	0.9425	0.7980	0.8564
Baseline		**0.8469**	0.1396	0.2350	**1.0000**	0.1238	0.2153

Table 2 shows the average detection performance of each approach in the five-fold cross validation. From this table we can observe the detection performance of ME-3D-G dramatically outperforms the baseline with respect to most metrics (four out of all six metrics) no matter we adopt the under-sampling or over-sampling approach. Although the baseline slightly outperforms ME-3D-G with respect to *Recall* and $Recall_{UX}$, its *Precision* and $Precision_{UX}$ are too low (i.e., 0.1396 and 0.1238) to be applied for real applications.

4.5 The Efficiency of the Model

Since the training of a detection model can be conducted in a server, we mainly concern the detection efficiency and memory cost of the detection model. Indeed, all MaxEnt based detection models discussed above are very efficient since the inference process of MaxEnt is very simple and all used contextual features are easy to extract. To be specific, in our experiments, the average time of making a detection for ME-3D-G, ME-3D, and ME-G are 20.1us, 18.2us, and 17.4us, respectively. Moreover, the memory costs of these MaxEnt based detection models are 4.57M, 4.57M, and 4.56M, respectively, which are much less than the memory limit of most modern smart phones.

5 Concluding Remarks

In this paper, we studied the problem of detecting subway arrivals for passengers in the train, which can enable a wide range of potential applications and services, such as subway arrival reminding services. The key idea of our approach is to collectively combine the evidences from multiple contextual information, which is collected by various sensors in mobile devices.

As illustrated in our experiments, for a subway arrival detection model, the extendability is critical for its success in practical applications. A training data set which reflects more common properties of the contextual data from all subway lines and less special properties of the contextual data from some particular subway lines may help to improve the extendability of a detection model. As for our future work, we plan to investigate some effective methods of selecting such "common" subway contextual data.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China Grant Nos. 61101133 and 61173158.

References

1. Android APIs (2013), `http://developer.android.com/reference/packages.html` (accessed June 22, 2013)
2. Deblauwe, N., Ruppel, P.: Combining gps and gsm cell-id positioning for proactive location-based services. In: MobiQuitous 2007 (2007)
3. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl
4. Liu, H., Darabi, H., Banerijee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews
5. Maloof, M.A.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II (2003)
6. Malouf, R.: A comparison of algorithms for maximum entropy parameter estimation. In: COLING 2002 (2002)
7. Paek, J., Kim, K.-H., Singh, J.P., Govindan, R.: Energy-efficient positioning for smartphones using cell-id sequence matching. In: MobiSys 2011 (2011)
8. Reddy, S., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Determining transportation mode on mobile phones. In: ISWC 2008 (2008)
9. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using mobile phones to determine transportation modes. ACM Trans. Sen. Netw.
10. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn.
11. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explor. Newsl
12. Wikipedia. 4G — Wikipedia, the free encyclopedia (2013), `http://en.wikipedia.org/wiki/4G` (accessed June 22, 2013)
13. Wikipedia. Beijing Subway — Wikipedia, the free encyclopedia (2013), `http://en.wikipedia.org/wiki/Beijing_Subway` (accessed June 22, 2013)

Vision Based Multi-pedestrian Tracking Using Adaptive Detection and Clustering

Zhibo Yang and Bo Yuan

Intelligent Computing Lab., Division of Informatics
Graduate School at Shenzhen, Tsinghua University
Shenzhen 518055, P.R. China
yangzhibo450@gmail.com, yuanb@sz.tsinghua.edu.cn

Abstract. This paper proposes a novel vision based multi-pedestrian tracking scheme in crowded scenes, which are very common in real-world applications. The major challenge of the multi-pedestrian tracking problem comes from complicated occlusions, cluttered or even changing background. We address these issues by creatively combining state-of-the-art pedestrian detectors and clustering algorithms. The core idea of our method lies in the integration of local information provided by pedestrian detector and global evidence produced by cluster analysis. A prediction algorithm is proposed to return the possible locations of missed target in offline detection, which will be re-detected by online detectors. The pedestrian detector in use is an online adaptive detector mainly based on texture features, which can be replaced by more advanced ones if necessary. The effectiveness of the proposed tracking scheme is validated on a real-world scenario and shows satisfactory performance.

Keywords: Pedestrian Tracking, Detection, Crowded Scene, Clustering.

1 Introduction

In recent years, vision based multi-pedestrian tracking has received growing interests from the computer vision community. Its typical applications include railway transport security, pedestrian traffic management, detection of overcrowded situations and people number counting. However, multi-pedestrian tracking in complicated environments still remains as a challenging task due to factors such as frame loss, occlusions, unusual appearances etc.

Significant efforts have been devoted to the detection and tracking of individuals in groups [1-3]. In general, the number of objects to be identified is small and the shape segmentation methods in use cannot separate individuals well in very crowded scenes (e.g., Fig. 1). It is also possible to only employ the tracking module, which may be based on sampling and particle filtering [4], MCMC [5] or greedy dynamic programming [6]. The tracking performance depends on the initial state provided by other detectors and none of these methods can guarantee reliable performance on overlapping targets. Furthermore, except the study based on linear programming [7], these approaches often perform badly when targets appear discontinuously in the video.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 58–66, 2013.

Fig. 1. Examples of crowded scenes (left: cross road; right: company main entrance)

To handle occlusions, Sidla et al. [8] used active shape models and the KLT (Kanade-Lucas-Tomasi) tracking algorithm to compute regions of interest and linked new shape locations to their corresponding trajectories. Lin et al. [9] estimated the number of people in crowded scenes using perspective transforms. Their approaches are designed for counting purpose and cannot track pedestrians in real time. With the multi-camera approach [10, 11], overlapping people can be tracked by different cameras and individual trajectories are processed separately over long sequences. In this paper, the objective is to realize multiple tracking with a single camera.

Inspired by the idea of data association [12] and adaptive detection [13], we decompose multiple objects tracking into three steps. In the first step, a detector trained in an offline manner estimates the number and locations of targets. Pedestrian detection has been extensively studied and a range of ready-to-use detectors are available. Previous studies have shown clearly that tracking algorithms based on whole body detection using shape features [8] are limited in their performance as soon as pedestrians are partially occluded. We adopt the idea proposed in [14] but use Haar-like features to train our two-part body detectors to solve the issue of partial occlusion. The second step considers the most common problem in practice where the detector produces false-negative results. A target validation module is used to predict the possible location of the missing target and redetect the target in the possible positions by template matching. In the third step, target locations and time information are combined to form trajectories using sequential leader clustering (SLC). The outliers (false-positive results) can be easily detected by this method.

The rest of the paper is organized as follows: Section 2 introduces the framework of our approach and presents some basic modules such as AdaBoost for training the pedestrian detector and sequential leader clustering. The key algorithms are discussed in Section 3 and 4. The experiment results will present in Section 5 and this paper is concluded in Section 6 with some discussion and a list of directions for future work.

2 Framework

The general framework of our approach is shown in Fig. 2. We use a state-of-the-art pedestrian detection method [14] based on body parts representation. The detection module consists of two stages: detection of body parts and combination of body parts. The major advantage of this approach is that it can deal with partial occlusions

(e.g., when the lower part of an individual is occluded, the upper part may still be detected). The human body is divided into two parts: head-shoulder (Ω-like part) and leg (Π-like part). Haar-like features are extracted from the training set containing both pedestrians and non-pedestrians samples and two classifiers are trained to classify these two parts based on AdaBoost, which selects a small number of critical Haar-like features including edge features, line features and center-surround features.

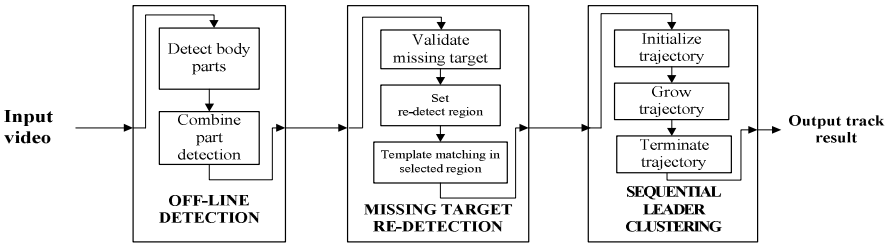


Fig. 2. A schematic framework of the proposed pedestrian tracking system

Regardless of the specific detector in use, the missing target validation is indispensible. It is very common that an individual may fail to be detected in a certain frame but is recognizable in previous and following ones. These false negative results are often isolated in time and should be re-detected (the second step in Fig. 2). How to confirm whether a false negative detection (a target is missing) or a false positive detection exists is an interesting issue. We model the belonging likelihood of new observations based on spatial, motion and appearance consistencies. Once new detections violate the constraints constructed by the likelihood model, the tracking system will indicate the missing targets and set the search area predicted by our model.

Note that the detector itself does not distinguish pedestrians and we propose to use clustering such as SLC to distinguish people from each other. Each pedestrian at each time step is specified by a data point and clustering is used to form a continuous trajectory for each pedestrian. To the best of our knowledge, this is the first time that SLC is employed to solve complex tracking problems.

3 Missing Target Re-detection

3.1 Validation of Detection Results

For a fixed position camera, a target's movement in the monitor region has to obey certain rules. Moving objects can only enter or leave the screen from the edge line rather than suddenly appear at the center of the screen. In general, a pedestrian walks at a speed of 5-6 km/h. The appearance especially the size of a target changes little within limited time frames and we aim to get a sequence of rectangular boxes for each detection result. However, a detector is usually not sufficient for this purpose and a

validation module is required to find the false detections that violate the above rules. The observed state P_t^i of the i^{th} detected pedestrian at time t is defined as:

$$P_t^i = \{c_t^i, s_t^i, v_t^i\}, \quad c = (x, y), \quad s = (w, h), \quad v = (dx, dy) \tag{1}$$

Where c is the coordinate of the rectangle's center, s is the width and the height of the rectangle box and v is the velocity of the moving rectangle. To validate the false negative detection and identify overlapping situations, new detections are compared with historic detections using similarity measurement. Three distance functions are proposed to measure the similarity between two rectangles (Eq. 2). D_1: *spatial proximity*. The spatial proximity describes the consecutive movement of the same person, measured by the Euclidean distance between locations. D_2: *box size*. Different pedestrians are unlikely to have identical box size and D_2 is defined as the normalized difference between pedestrian sizes. D_3: *velocity coherence*. The velocity vectors contain the direction information that is important for dealing with occlusions.

$$D_1 = \|c_m - c_n\|, \quad D_2 = \|s_m - s_n\|, \quad D_3 = \|v_m - v_n\| \tag{2}$$

Missing targets: when new targets are detected at time t, historic detections have been labeled using SLC. After comparing all m new rectangles with the last few rectangles in the trajectory of the i^{th} pedestrian, if the condition specified by Eq. 3 is met (T is threshold), it is assumed that the target corresponding to the i^{th} pedestrian is missing and re-detection is required in its neighborhood.

$$D_1^{i,j} > T_1, and \quad D_2^{i,j} > T_2, and \quad D_3^{i,j} > T_3, where \quad j = (0 \cdots m) \tag{3}$$

Full occlusions: the employed detector can solve most partially occlusions and we only need to focus on totally occlusions. In the simplest situations, the i^{th} pedestrian stands between the j^{th} pedestrian and the camera. In this case, there may be a detection n that satisfies the conditions in Eq. 4 and a flag will be set so that the subsequent clustering module (Section 4) can correctly handle this occlusion. It is easy to extend the two pedestrian occlusions to multi-pedestrian occlusions, and we duplicate the observation the times same to the number of being occluded pedestrians.

$$D_1^{k,m} < T_1, and \quad D_2^{k,m} < T_2, and \quad D_3^{k,m} < T_3, where \quad k = (i, j \cdots) \tag{4}$$

3.2 Re-detection

Once missing pedestrians are identified by the method specified in Section 3.1, a search process is carried out within a limited region based on the spatial and velocity information instead of the entire image to reduce the computational cost. Suppose that the missing target's last state is P_t, center at (x_t, y_t), size (w_t, h_t) and the average

velocity of this target's movement in one step is (*dx, dy*). The most likely center of the missing target is set to (x_t+*dx*, y_t+*dy*). An empirical re-detection region is described by Eq. 5.

$$region:\{x_t + dx \pm 3w_t,\ y_t + dy \pm 3h_t\} \tag{5}$$

Next, the last two detections of the pedestrian are stored in the form of color histogram and a template matching method [15] is used to traverse the re-detection region to identify the missing target.

4 SLC for Multiple Tracking

Fig. 3(left) shows a scenario where the camera is equipped on top of the main entrance of an office building. In Fig. 3(middle), the x-axis and y-axis represent the location of pedestrian and the vertical axis represents the time. A trajectory is created by sequentially adding pedestrian records into the cluster corresponding to the same pedestrian. Fig. 3(right) shows an example where there are three trajectories (clusters) represented by red, blue and green respectively and a new record (black cross) to be identified. In this paper, we use SLC as the clustering technique, which measures the similarity between the new record and each of the three clusters and assign it to the most similar cluster (trajectory).

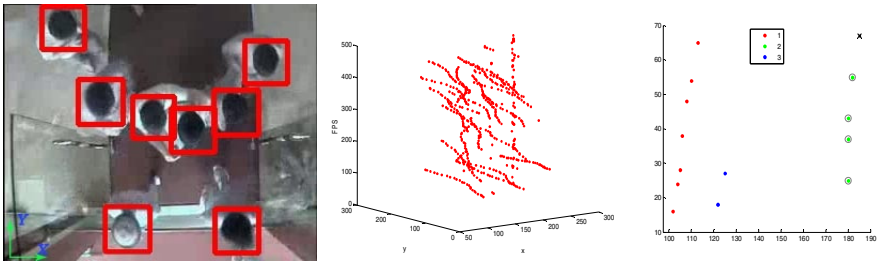


Fig. 3. A real-world scenario where the camera is equipped on top of the entrance (left), the 3D representation of pedestrian records (middle) and trajectory expansion (right)

SLC is used to cluster the detections into a number of clusters in an incremental manner, corresponding to the trajectories of unique pedestrians (Algorithm 1). Some modification is made to the original SLC so that whenever a full occlusion is detected (Eq.4, Section 3.1), duplicates of the original detection will be created with each one assigned to one of the intersecting trajectories.

Algorithm 1. SLC for Multiple Tracking

dist(x,y)—function for measuring similarity between objects x and y
createCluster—function for creating a cluster that includes one element x
addToCluster—procedure for adding element x to cluster c
selectFrom— procedure that selects an element from clusters
e—threshold of closeness of exemplar
n—the total detections in time t
bigNum—the threshold for creating a new cluster
cm—cluster with mark bit c_flag in the end
c_flag—a flag for indicating a cluster cm has receive a new element

```
SLCinTracking(x[i], clusters, e)
for (i=0; i<n; i++)
{
    If empty(clusters) then
            clusters = {createCluster(x[i])};
          return;
    dmin=bigNum;
       cm=selectFrom(clusters);
    for c∈ clusters &c_flag≠1 do
        dd=dist(x[i],c);
        ifdmin>dd&dd≦e then
        dmin=dist(x,c); cm=c;
    if dmin<bigNum then
        addToCluster(x[i],cm); c_flag=1;
    else
    createCluster(x[i]);
}
```

5 Experiment

5.1 Standard Benchmark Results

In this section, we present the results of our method on the CAVIAR dataset [16]. This dataset is very challenging due to sever occlusions and cluttered and complicated backgrounds. The CAVIAR dataset contains 26 video sequences of a walkway in a shopping center taken by a single fixed camera with frame size 385×288 and frame rate 25fps. For comparison purpose, the results from some previous studies [12, 14] are also included. We evaluated the tracking performance from three aspects: completely tracked pedestrians (CT), completely lost pedestrians (CL), and partially tracked pedestrians (PT). The experiment results are shown in Table 1, where GT represents the ground truth.

Table 1. Experiment results on CAVIAR

Method	GT	CT	PT	CL	CT RATE
Wu et al. [14]	140	106	25	9	75.17%
Zhang et al. [12]	140	104	29	7	74.28%
Our Method	140	113	19	8	80.72%

We can see that our method achieved a higher CT rate, capable of fully tracking more pedestrians than existing methods. This is mainly due to introduction of the re-detection mechanism. Fig. 4 shows some of the recognition examples.

Fig. 4. Examples of the tracking results on the CAVIAR dataset

5.2 Surveillance Applications

We applied the proposed method to a real-world application on pedestrian counting based on the video from the CCTV monitoring a large company's main entrance. Compared to the CAVIAR dataset, the back ground was much cleaner, with less number of occlusion situations due to position of the camera but the people were more crowded in the video (Fig. 5). The frame size was 320×240 and frame rate was 15fps. A large set of positive and negative samples were collected for training the detector. The overall precision was more than 96% and the processing time for each frame was around 50ms on a 2G Hz dual core computer with 2G memory.

Fig. 5. Examples of the pedestrian counting application

6 Conclusions

In this paper, we applied the sequential leader clustering method to multiple pedestrian tracking, which provides a new direction for the research in this area and is much simpler than other existing techniques. The proposed method can also validate

the detection results and, if necessary, re-detect the missing targets at a very low cost. The experiment results on a standard benchmark video dataset and a real-world application show that the precision of our method is satisfactory in most scenes with fast processing speed. In the future, we will focus on investigating more advanced pedestrian detectors, incorporating more cues in validation and testing other clustering or classifying methods to further improve the tracking performance.

Acknowledgement. This work was partially conducted while the first author was working at the R&D department of MICROPROFIT Company. We are also grateful to Mr. Haoran Xin for his support during the test of the prototype system.

References

1. Zhao, T., Nevatia, R.: Bayesian Human Segmentation in Crowded Situations. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 459–466 (2003)
2. Rittscher, J., Tu, P.H., Krahnstoever, N.: Simultaneous Estimation of Segmentation and Shape. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 486–493 (2005)
3. Leibe, B., Seemann, E., Schiele, B.: Pedestrian Detection in Crowded Scenes. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 878–885 (2005)
4. Chang, C., Ansari, R., Khokhar, A.: Multiple Objects Tracking with Kernel Particle Filter. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 566–573 (2005)
5. Ge, W., Collins, R.T.: Multi-Target Data Association by Tracklets with Unsupervised Parameter Estimation. In: 19th British Machine Vision Conference, pp. 93.1–93.10 (2008)
6. Shafique, K., Shah, M.: A Noniterative Greedy Algorithm for Multiframe Point Correspondence. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(1), 51–65 (2005)
7. Jiang, H., Fels, F., Little, J.J.: A Linear Programming Approach for Multiple Object Tracking. In: 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
8. Sidla, O., Lypetskyy, Y., Brandle, N., Seer, S.: Pedestrian Detection and Tracking for Counting Applications in Crowded Situations. In: 2006 IEEE International Conference on Video and Signal Based Surveillance, p. 70 (2006)
9. Lin, S.F., Chen, J.Y., Chao, H.X.: Estimation of Number of People in Crowded Scenes Using Perspective Transformation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 31(6), 645–654 (2001)
10. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera People Tracking with A Probabilistic Occupancy Map. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 267–282 (2008)
11. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple Object Tracking Using K-shortest Paths Optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(9), 1806–1819 (2011)

12. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-object Tracking Using Network flows. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Face-tld: Tracking-Learning-Detection Applied to Faces. In: 2010 IEEE International Conference on Image Processing, pp. 3789–3792 (2010)
14. Wu, B., Nevatia, R.: Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors. International Journal of Computer Vision 75(2), 247–266 (2007)
15. Pereira, S., Pun, T.: Robust Template Matching for Affine Resistant Image Watermarks. In: 2000 IEEE International Conference on Image Processing, pp. 1123–1129 (2000)
16. CAVIAR Dataset, `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1`

Drilling Cost Prediction Based on Self-adaptive Differential Evolution and Support Vector Regression

Huaxian Pan[1], Guojian Cheng[2], and Jian Ding[3]

[1] Xingzhi College, Xi'an University of Finace and Economics, Xi'an 710038, Shaanxi, China
panhuaxian@gmail.com
[2] School of Computer Science, Xi'an Shiyou University, Xi'an 710065, Shaanxi, China
[3] Xinjiang Oil Field Company Shixi Field Operational Zone, Karamay, 834000, Xinjiang, China

Abstract. Prediction of drilling cost in oil and gas field has an important impact on correct economic decision-making and economic efficiency of oilfield enterprises. Support Vector Machine (SVR) is used to predict the drilling cost of oil and gas in this paper. In order to overcome problem of easy to cause local optimum of SVR, this paper uses self-adaptive differential evolution (SaDE) to train SVR to accelerate the speed of parameters optimization and improve the predictive accuracy of the model. The proposed model is applied to predict the drilling costs in one of the Chinese oil company. SaDE is also compared with three parameter optimization methods, Differential Evolution(DE), Grid Search(GS) and Genetic Algorithms(GA). The experimental results show that, SaDE-SVR model is better than DE-SVR, GS-SVR and GA-SVR in terms of accuracy and convergence speed in predicting the drilling cost. It validates the effectiveness of SaDE-SVR applied in prediction of drilling cost of oil and gas field.

Keywords: Drilling cost, SVR, SaDE, Oil and gas.

1 Introduction

The oil and gas drilling operation is the main job of a series of drilling activities. Drilling cost accounts for a large proportion of the entire wellbore job. Therefore, to study the drilling cost makes a significant contribution to analyze the comprehensive drilling cost. Drilling cycle, the type of wells, wells location and drilling footage are major factors that affecting drilling cost of oil and gas according to a large number of previous studies [1].

Methods for drilling cost prediction include: linear regression, nonlinear regression, chaos forecasting and trend forecasting. Zhao Yue et al. presents a BP neural network model to predict the drilling cost [2]. However, there are significant drawbacks of neural network, such as the network structure is difficult to determine. Currently, SVR is commonly used to solve the problems of

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 67–75, 2013.

small sample size, nonlinear and high dimensional pattern recognition. SVR has many advantages, the global optimal solution can be obtained in theory, the model structure is simple, and the actual problem is transformed into a high-dimensional feature space by nonlinear transformation. Some studies that predict drilling cost of oil and gas based on SVR, example includes Ma Jiachuan to verify the effectiveness of the algorithm of SVR applied in drilling cost prediction [3]. The past study found that the parameters of SVR are the key factors that affect the learning performance of the algorithm. They have an important impact on the learning performance, generalization ability and the risk of the model. It has important theoretical and practical significance for SVR development when studies the parameter optimization methods.

DE is one of the most popular heuristic methods for solving single-objective optimization problems in continuous search space[4]. It is a relatively new evolutionary algorithm, and has outstanding performance in the areas of function optimization. However, this algorithm has the disadvantages of optimizing in long time and premature convergence. The trial-and-error method is used to determine the control parameters of DE for dealing with different problems. It consumes a large amount of computing time. An adaptive method is introduced to adjust the control parameters of DE, in order to improve the searching speed, overcome premature convergence problem of heuristic algorithms, and improve the global convergence of the algorithm. Therefore, a SaDE-SVR model is constructed for prediction of drilling cost to verify the effectiveness of the algorithm and test the predictive performance of the proposed model. The parameter optimization method of SaDE is also compared with GS, DE and GA to verify the effectiveness and practicality of SaDE-SVR model applied in predicting drilling cost of oil and gas.

2 SaDE-SVR Algorithm

To solve the general optimization problem, DE algorithm can be expressed as follows[5]:

For the objective function $f : x \subseteq \Re^D \rightarrow \Re$ where $X \neq \phi$, the minimization problem is to find:

$$x^* \in X, f(x^*) \leq f(x) \ \forall x \in X \tag{1}$$

where $f(x^*) \neq -\infty$.

2.1 Initialization

Let X_j^p is the jth chromosome of generation p[6],

$$X_j^p = [x_{j,1}^p, x_{j,2}^p, ..., x_{j,i}^p]^T, \ j \in \{1, 2, 3\}, p = 1, 2, ...p_{max} \tag{2}$$

where $x_{j,i}^p$ represents the ith parameter of jth individual in generation p, p_{max} is the maximum evolution generation. Therefore, X_j^p is a candidate solution of the objective function.

The individual $x_{j,i}^0$ in the initial population is set as a floating-point number which is even random distribution in the upper and lower bounds.

$$x_{j,i}^0 = rand_{j,i}(0,1)(x_{j,i}^U - x_{j,i}^L) + x_{j,i}^L \; j \in 1,2,...NP; i = (1,2,3) \tag{3}$$

where $x_{j,i}^U$,$x_{j,i}^L$ are the upper and lower boundaries values of penalty coefficient C, insensitive loss function ε, and parameter of kernel function γ respectively. $rand(0,1)$ represents a random variable within the range [0,1].

2.2 Fitness Function

The corresponding parameter of each individual vector in the population determines the fitness function $fit(C,\gamma,\varepsilon)$, it is calculated as follows for a given chromosome X_j^p.

$$fit(X_j^p) = \frac{1}{n}\sum_{i=1}^{n}(t_i - f_i)^2 \tag{4}$$

where t_i is the actual value of experimental data, f_i is the predicted value of SVR, and n is the total number of forecasting samples.

2.3 Muation

The four most frequently used mutation strategies are listed as follows:

DE/rand/1

$$V_j^p = X_{r_1^j}^p + F \cdot (X_{r_2^j}^p - X_{r_3^j}^p) \tag{5}$$

DE/rand-to-best/2

$$V_j^p = X_{r_1^j}^p + F \cdot (X_{best}^p - X_{r_1^j}^p) + F \cdot (X_{r_2^j}^p - X_{r_3^j}^p) + F \cdot (X_{r_4^j}^p - X_{r_5^j}^p) \tag{6}$$

DE/rand/2

$$V_j^p = X_{r_1^j}^p + F \cdot (X_{r_2^j}^p - X_{r_3^j}^p) + F \cdot (X_{r_4^j}^p - X_{r_5^j}^p) \tag{7}$$

DE/current-to-rand/1

$$V_j^p = X_j^p + K \cdot (X_{r_1^j}^p - X_j^p) + F \cdot (X_{r_2^j}^p - X_{r_3^j}^p) \tag{8}$$

Let X_j^p is the current evolution individual, randomly select five population vectors $X_{r_1^j}^p, X_{r_2^j}^p, X_{r_3^j}^p, X_{r_4^j}^p, X_{r_5^j}^p$ in the current population, they are the individuals different form individual X_j^p, and $r_1^j, r_2^j, r_3^j, r_4^j, r_5^j$ are mutually exclusive integers randomly generated within the range $[1, NP]$. And X_{best}^p is the best

individual vector with the best fitness value in the population at generation p. K within the range $[0, 1]$ is the control parameter. The positive amplification factor within the range is used to control the scaling of the difference vectors. The positive scale factor F within the range [0,2] is used to control the scaling of difference vectors, its size is closely related to the convergence speed of the algorithm[7].

2.4 Crossover

The current evolution individual X_j^p is discrete crossed with the individual V_j^p after mutation, generating the candidate individual vector U_j^p:

$$u_{j,i}^p = \begin{cases} V_{j,i}^p & if\ rand(i) \le CR\ or\ (i = i_{rand}) \\ x_{j,i}^p & Otherwise \end{cases} \quad i \in 1, 2, 3, CR \in [0, 1] \tag{9}$$

where $u_{j,i}^p$ is a trial vector in the next generation and it is the parameter of U_j^p after crossover. $V_{j,i}^p$ is the corresponding parameter of mutation individual V_j^p, $rand(i)$is random number within $[0, 1]$. CR within the range $[0, 1]$ is the crossover rate to control the fraction of the parameter values, i_{rand} is a randomly chosen integer in the range $[1, 3]$ to ensure that there exist at least one parameter in $u_{j,i}$ differing from the target vector.

We firstly introduced a fixed number of iterations as the learning period LP, and the $P_{l,G}$ is defined as probabilities of choosing different strategies will be updated at each subsequent generation, which is updated based on the success and failure memories. So when $LP < G$, the probability of choosing the lth $(l = 1, 2, 3, 4)$ strategy is updated by [8],

$$P_{l,G} = \frac{S_{l,G}}{\sum_{l=1}^{4} S_{l,G}} \tag{10}$$

where

$$S_{l,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{l,g}}{\sum_{g=G-LP}^{G-1} ns_{l,g} + \sum_{g=G-LP}^{G-1} nf_{l,g}} + \varepsilon \ (l = 1, 2, 3, 4) \tag{11}$$

$ns_{l,g}$ is the number of trail vectors generated by the lth strategy that can successfully go into the next generation. Conversely, $nf_{l,g}$ is the number of trail vectors that are discarded in the next generation. The small constant value is used to avoid the possible null success rates.

As refer to other parameters, appropriate values of F and CR generally fall into a small range for a given problem. So a set of F and CR are randomly generated according to $N(0.5, 0.3)$ and $N(0.5, 0.1)$.

2.5 Selection

The selection operation can be expressed as follows:

$$X_j^{p+1} = \begin{cases} X_j^p & if\ f(U_j^p) > f(X_j^p) \\ U_j^p & Otherwise \end{cases} \tag{12}$$

The selection operation ensures the individuals in the next generation are better than or at least similar to the individuals of the previous generation[8]. The mutation, crossover and selection operation are repeated until the maximum learning iterations are completed or the goal is reached.

3 Experiments

The drilling cycle, type of wells, wells location and drilling footage are selected as the parameters for prediction of drilling cost of oil and gas field in this paper. There are some non-quantifiable factors, e.g. the type of wells and wells location, and the values of four parameters are in different ranges. If the values are not limited to a specific range, SVR will do not converge in the training process, or it will produce meaningless results. Therefore, non-quantitative factors must be quantified. The type of wells is generally divided into oil production wells, gas production wells, oil exploration wells and gas exploration wells. The type of wells and the wells location are expressed in integer between $[1, 4]$ and $[1, 14]$ respectively. And all samples used in the experiment must be normalized within the range $[0, 1]$ through Equ.13 before training,

$$\acute{x}_i = \frac{X_i - \min\limits_{1 \le i \le n}(x_i)}{\max\limits_{1 \le i \le n}(x_i) - \min\limits_{1 \le i \le n}(x_i)} \tag{13}$$

Where $min(x_i)$ and $max(x_i)$ represent the minimum and maximum values of the input vector X respectively. In order to evaluate the performance of the proposed method applied in drilling cost prediction, drilling cost data of different blocks in a drilling engineering Corporation in 2008 is used in the experiments. The drilling cost data of 1614 wells in 14 district are selected as the experimental data, a total of 1594 samplings, which is divided into training dataset and testing dataset, are used in the SaDE-SVR models. 1564 data points are used to build the SaDE-SVR learning model to obtain the optimal parameters of SVR, and 30 data points for testing.

The DE-SVR, GA-SVR and GS-SVR models are compared with the proposed method in order to evaluate the effectiveness of SaDE-SVR for prediction of drilling cost in this paper. These methods use the same dataset to compare the accuracy in the experiment. And 5-fold cross-validation method is applied in all the models. The radial basis function(RBF) is one of the most commonly used kernel function in SVR technique. Hence, RBF is used for SVR training in this study. The RBF requires setting of the parameter γ in addition to the regular parameters C and ε. Tab.1 lists some initial values of parameters of SVR and SaDE for training.

The optimal parameters of γ, ε and C is searched by SaDE. We use the relative error (RE) and correlation coefficient (R) to evaluate the performance of the proposed method in the experiment. We select the same search area, i.e. the same values range of C, γ, ε the number of iterations and population size are fixed to 20, to maintain approximate continuous searching time for all methods

Table 1. Parts of parameter settings of SVR and SaDE

Parameters	RBF			Maximum iterations	Inital population size
	C	γ	ε		
Parameter value / range	[0.1,100]	[0.1,10]	[0.01,1]	20	20

in order to get a fair result. The prediction results of some samples are shown in Tab.2, RE-DE, RE-GA, RE-GS and RE-SaDE represent the relative error of the prediction results of DE, GA, GS and SaDE respectively. It can be seen from the table, the prediction accuracy of SaDE-SVR is higher than GA and GS, and the predicted results are approximate to the traditional DE.

Table 2. Comparison of the test results of DE,GA,GS and SaDE

Samples	Type of wells	Wells location	Drilling cycle (/month)	Drilling footage (/m)	Drilling cost (/yuan)	RE-DE	RE-GA	RE-GS	RE-SaDE
1	1.00	3.00	0.28	2,085.00	1,338,649.59	6.59	11.00	4.94	6.61
2	1.00	7.00	0.19	2,099.00	1,621,499.37	5.79	11.54	3.84	5.60
3	2.00	8.00	0.51	3,453.00	1,690,188.86	6.43	17.78	5.18	6.33
4	1.00	1.00	0.19	1,924.00	1,067,889.62	11.40	28.18	10.98	11.01
5	1.00	2.00	0.35	2,185.00	1,127,038.19	1.36	13.72	5.14	2.37

Fig.1 reports the comparison of actual values and the corresponding predicted drilling cost of DE-SVR, GA-SVR, GS-SVR and SaDE-SVR model. The figure shows that, the predicted value of DE-SVR and SaDE-SVR model is very close to the actual value. This indicates that the performance of DE and SaDE is better than Genetic Algorithms and Grid Search algorithm. DE and SaDE have the advantages of good reliability, high accuracy and can be effectively used in prediction of drilling cost.

Tab.3 lists the comparison of maximum, minimum and average relative error value of the four methods. There is an obvious deference in terms of minimum relative error, maximum relative error and the average relative error of SaDE-SVR, DE-SVR, GS-SVR and GA-SVR. From the results, we can observe that the average relative error of the predicted value and the true value of the four models are 4.09%, 4.18%, 4.41% and 15.8% respectively. The accuracy of DE-SVR and SaDE-SVR are better than GS-SVR and GA-SVR, the maximum relative error of the proposed model is only 10.93%, the minimum relative error is 0.77%.

Fig.2 shows that, the deviation of predictive value in DE-SVR and SaDE-SVR are very small, some experiments even have no deviation. The regression coefficient of DE-SVR and SaDE-SVR are relatively high ($R = 0.9227$ and $R = 0.9398$, respectively). While in GS-SVR model, $R = 0.9031$, and GA-SVR is the worst, the predicted results is significantly deviate from the actual value

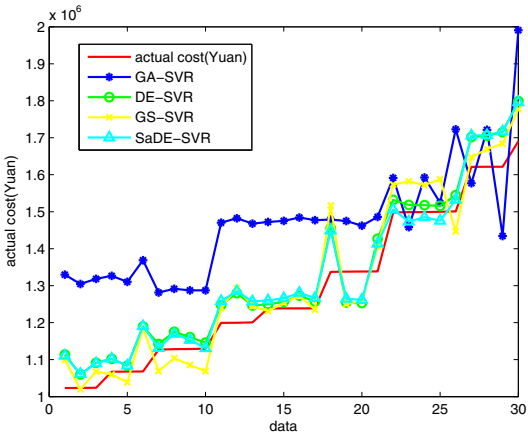


Fig. 1. Comparison of the predicted values with actual values

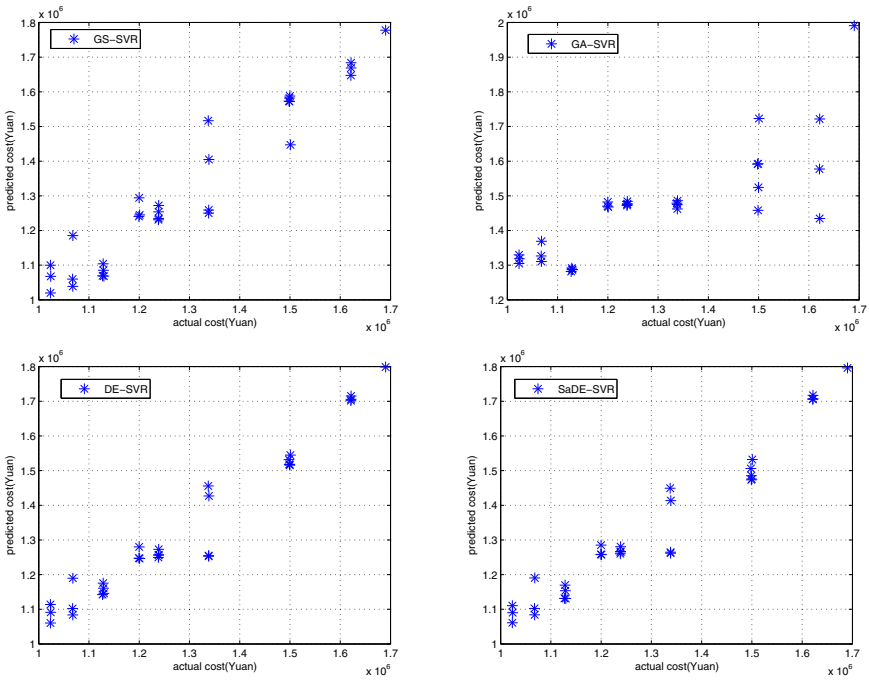


Fig. 2. Cross plot of the predicted values and actual values

($R = 0.7336$). These results demonstrate the robustness of the proposed method. The experimental results illustrate that the prediction of DE and SaDE is more accurate than GA and GS.

Table 3. Comparison of the predicted values

Relative Error	Methods	Values
Minimum	SaDE-SVR	0.77
	DE-SVR	0.93
	GA-SVR	1.64
	GS-SVR	0.33
Maximum	SaDE-SVR	10.93
	DE-SVR	11.4
	GA-SVR	29.94
	GS-SVR	13.41
Average	SaDE-SVR	4.09
	DE-SVR	4.18
	GA-SVR	15.8
	GS-SVR	4.41

The parameters of SVR after searched by four models, and the fitness values with the change of iterations are shown in Fig.3. The figure shows that convergence speed of DE-SVR, GA-SVR and SaDE-SVR is significantly faster than GS-SVR. DE-SVR and SaDE-SVR model converge simultaneously after $5th$ iteration, and SaDE-SVR converges earlier than DE-SVR before $5th$ iteration, which verify the advantage of SaDE in convergence speed. According to the results of the analysis in this paper, the proposed algorithm has the following advantages compared to DE: fast convergence speed, less training time. SaDE has the advantages of good reliability, high accuracy and it can be effectively used in prediction of drilling cost.Therefore, SaDE-SVR model can be used to predict the drilling cost of oil and gas with relatively good predicted result.

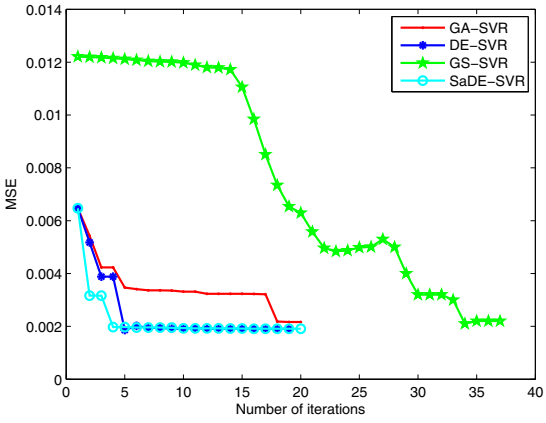


Fig. 3. Comparison of convergence of GA,DE, GS and SaDE

4 Conclusion

SVR with self-adaptive differential evolution is used to optimize the three parameters of SVR to predict the drilling cost. The traditional DE , GA and GS are compared with SaDE. It can be found that the parameters of SVR obtained by SaDE which are used to drilling cost prediction is better than DE-SVR, GS-SVR and GA-SVR in terms of convergence speed and prediction accuracy. Although SaDE-SVR model is similar to GA-SVR model in time-consuming, the prediction accuracy of SaDE-SVR is higher than GA-SVR, and it shows a good convergence capability. Therefore, the performance of SaDE is better than DE, GA and GS, which verifies its effectiveness in drilling cost prediction and has a higher prediction accuracy.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (40872087).

References

1. Fan, M., et al.: Study of cost drivers of oil and gas drilling operations. Journal of Southwest Petroleum University 1, 113–117 (2007)
2. Liu, T., et al.: Drilling cost projection of oil and gas based on BP neural network. Journal of Xian Shiyou University(Natural Science Edition) 1, 87–90 (2010)
3. Ma, J.: Research on and Application of SVR-based Well-drilling Cost. Xian Shiyou University 1–3 (2010)
4. Price, K.: Differential evolution: A fast and simple numerical optimizer. In: Proc. North American Fuzzy Information Processing Conf, pp. 524–527 (1997)
5. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation 13, 398–417 (2009)
6. Sanchez, A.D.: Advanced vector machines and kernel methods. J. Neurocomputing 1, 5–20 (2003)
7. Price, K., Storn, R.: Differential evolution. Dr. Dobbs J. 18–20 (1997)
8. Abbasss, H.A.: Self-adaptive pereto differential evolution. In: Proceedings of the IEEE 2002 Congress on Evolutionary Compution, pp. 831–836. IEEE Press (2002)

Web Service Evaluation Method Based on Time-aware Collaborative Filtering

Guisheng Yin, Xiaohui Cui, Hongbin Dong, and Yuxin Dong

College of Computer Science and Technology,
Harbin Engineering University, Harbin 150001, P.R. China
{yinguisheng,cuixiaohui,donghongbin,dongyuxin}@hrbeu.edu.cn

Abstract. Web service evaluation approaches predict the reliability of services running in the dynamic and unpredictable Internet environment, which helps the target users find the reliable services. Current approaches of evaluation based on the collaborative filtering lack the consideration of the time-aware effectiveness of the past data. To solve the problem, we proposed a web service evaluation approach based on the time-aware collaborative filtering, which computes the time-aware effectiveness of the past data and improves the process of the neighbor users finding. The real-world web service Qos dataset is conducted and the experimental results show that our approach achieves the lower prediction errors than other approaches.

Keywords: web service, time-aware effectiveness, evaluation prediction, collaborative filtering.

1 Introduction

In the dynamic and unpredictable Internet environment, web service evaluation approaches predict the reliable services for the target users. One way to implement the web service evaluation is using the objective past data to filter out the neighbor users with the same network environments as the target users, which collaboratively predict the reliability of the services that are not invoked by the target users [1].

A few approaches of the web service evaluation based on the collaborative filtering have been proposed, and the analysis of the prediction error of these approaches is carried on Movielens [2], for example Rong [3]. When there is a lack of the public dataset to test the performance of the web service evaluation approaches, Movielens is the only choice to some extents. However, the film ratings of Movielens are quite different from the reliability data of the web services, so the lack of the public dataset restricts the studies of the web service evaluation. Due to the above needs of the public dataset, [4] proposes a real-world web service Qos dataset WSDREAM [5]. WSDREAM records the throughput and time-out values of the services invoking. WSDREAM is still improving so far.

Generally, the recent data possess more effective than the outdated data when prediction. This idea of the time-aware effectiveness of the past data has been

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 76–84, 2013.

wildly used in other fields. For instance, the scholars of Trustworthy Software present DyTrust [6] and SerTrust [7], which employ the exponential decay reduction; The researchers [8] of Machine Learning deem the reduction of the time-aware effectiveness as a linear decay; The researches [9] of Concept Drift adopt the forgetting curve to describe the time-aware effectiveness. However, little research on the time-aware effectiveness of the past data has been done for the web service evaluation based on the collaborative filtering.

To address the problem of the time-aware effectiveness of the past data, we propose a time-aware framework of the web service evaluation. Our proposed framework computes the time-aware effectiveness of the past data, whose results are used to improve the neighbor users finding and reliability prediction.

The rest of the paper is organized as follows: Section 2 introduces the framework. Section 3 describes the quantification model of the time-aware effectiveness. Section 4 presents the improved neighbor users finding and reliability prediction. Section 5 describes the experiments. Section 6 concludes the paper.

2 Time-aware Evaluation Framework

FWSE-TCF(Framework of Web Service Evaluation based on Time-aware Collaborative Filtering) implements the processes of our approach, which is designed as a four-phase processes as shown in Fig. 1.

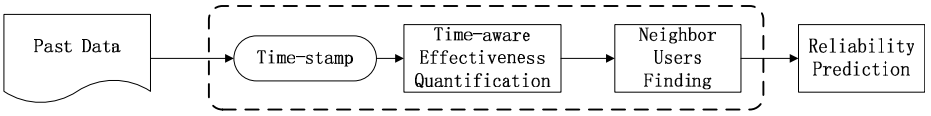


Fig. 1. Processes of FWSE-TCF

- In phase 1, we extract the time-stamps of the past data related with the target users.
- In phase 2, the extracted time-stamps are used by the quantification model of the time-aware effectiveness to evaluate the effectiveness of the past data.
- In phase 3, the neighbor users of the target users are identified. Then, Top-N is employed to filter out the N neighbor users.
- In phase 4, the reliability of the uninvoked services is predicted by the time-weighted past data observed by the N neighbor users.

Compared with the current approaches, the contributions of FWSE-TCF are the quantification model of the time-aware effectiveness of the past data and the improved neighbor users finding, which is marked as a dotted line in Fig. 1.

3 Time-aware Quantification

Compared with the outdated data, the data collected recently present more contribution for the web service evaluation. Therefore, we design a quantification model to evaluate the time-aware effectiveness of the past data.

Currently, the linear functions and the non-linear functions are two main function models to describe the decay of the time-aware effectiveness of the past data. The linear functions deem that the time-aware effectiveness reduces as a steady speed. The non-linear functions simulate the processes of the human forgetting, and the time-aware effectiveness reduces sharply at the beginning but gently at the end. Fig. 2 shows the reduction processes of the linear and the non-linear functions.

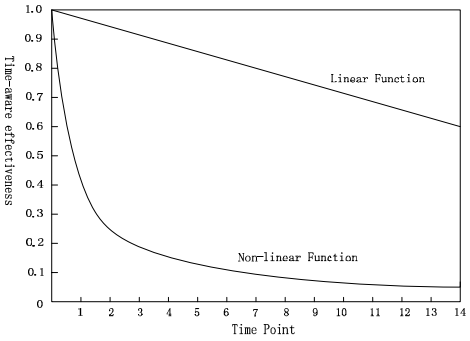


Fig. 2. Different reduction processes of the time-aware effectiveness

Current linear and non-linear functions are not suitable to quantify the time-aware effectiveness of the past data of the web service evaluation, for the reduction speed of the non-linear functions are too fast, while the linear functions are too slow. In fact, the decay of the time-aware effectiveness of the past data processes not only the linear reduction but also the non-linear reduction. Therefore, we can combine the linear and non-linear reduction to design a hybrid quantification model to evaluate the effectiveness of the past data. We adopt $p(h, t)$ to describe the time-aware effectiveness of the past data h at time t, which is defined as:

$$p(h,t) = f(t) = \{(T-t)/T\}^n (f_0 - f_T) + f_T. \tag{1}$$

where f_0 and f_T are the initial and final time-aware effectiveness. Assuming that the past data h is generate at time t_0 and T is the period of the time-aware effectiveness $p(h, t)$ reduction from f_0 to f_T. If $t - t_0 > T$, there is no need to compute the time-aware effectiveness by Eq.(1) and we directly use $f(t) = f_T$, which would alleviate the computation complex of the time-aware effectiveness and decrease the side effects of the outdated data to the web service evaluation.

In Eq.(1), n controls the reduction speed of the time-aware effectiveness over a period T. If we provide the fixed values for f_0, f_T and T, the reduction curves on different settings of n are shown in Fig.3. In Fig.3, n and T are the key parameters to control the reduction of the time-aware effectiveness. When $n = 1$, Eq.(1) becomes a linear reduction. When $n < 1$, Eq.(1) becomes a sort of non-linear convex functions. When $n > 1$, Eq.(1) becomes a sort of non-linear concave functions. To get reasonable settings of the parameters n and T, we resort to the experiments in Section 5.

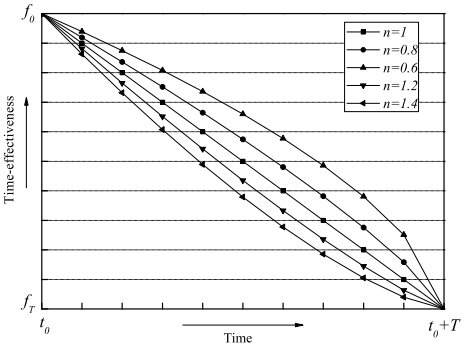


Fig. 3. The reductions of the time-effectiveness with different n

4 Neighbor User Finding and Reliability Prediction

4.1 Neighbor User Finding

In this section, we import the quantification results to improve the neighbor users finding. Assuming that u_i is the target user and $p(h,t)$ is the time-aware effectiveness of the past data h at time t, the improved similarity $sim(u_i, u_j, t)$ between u_i and the other user u_j is defined as:

$$sim(u_i, u_j) = \frac{\sum_{m \in W_{i,j}} (h_{i,m} p(h_{i,m}, t) - \bar{h}_i)(h_{j,m} p(h_{j,m}, t) - \bar{h}_j)}{\sqrt{\sum_{m \in W_{i,j}} (h_{i,m} p(h_{i,m}, t) - \bar{h}_i)^2 (h_{j,m} p(h_{j,m}, t) - \bar{h}_j)^2}} \quad (2)$$

where $W_{i,j}$ are the services commonly invoked by u_i and u_j, m is an arbitrary service in $W_{i,j}$, $h_{i,m}$ ($h_{j,m}$) is the past data related with service w_m invoked by u_i (u_j respectively), $p(h_{i,m}, t)$ ($p(h_{j,m}, t)$) is the time-aware effectiveness of $h_{i,m}$ ($h_{j,m}$ respectively) at time t, and $\bar{h}_i$ ($\bar{h}_j$) is the mean value of the past data of the service w_i (w_j respectively).

By Eq.(2), we can sort the neighbor users as the ascending order by $sim(u_i, u_j, t)$ and compose the final neighbor user sets $ord(u_i)$ by:

- Top-N. Selecting the N most similar users to compose the $ord(u_i)$.
- Threshold. Selecting u_j if the $sim(u_i, u_j, t)$ is more than a fixed threshold.

Top-N is better than threshold method for the web service evaluation. The reason is that threshold method is prone to cause the failure discovery of $ord(u_i)$. In Section 5, we would achieve the reasonable settings of the parameter N.

4.2 Reliability Prediction

Assuming that $W(ord(u_i))$ is composed by the services invoked by users from $ord(u_i)$ and w is a service from $W(ord(u_i))$, the reliability prediction of uninvoked services w for the target users is defined as:

$$h_{u_i,w}(t) = \bar{h}_{ui} + \sum\nolimits_{u \in ord(u_i)} \omega_u (h_{u,w} p(h_{u,w}, t) - \bar{h}_u). \tag{3}$$

where u is an arbitrary user in $ord(u_i)$, $\bar{h}_{u_i}$ ($\bar{h}_u$) is the mean value of the past data related with u_i (u respectively), $p(h_{u,w}, t)$ is the time-aware effectiveness of $h_{u,w}$ and ω_u is the similarity weight of u, which is defined as:

$$\omega_u = sim(u_i, u, t) / \sum\nolimits_{u_j \in \{ord(u_i) - u\}} sim(u_j, u, t). \tag{4}$$

where u_j is the other user in the set $\{ord(u_i) - u\}$ and $sim(u_i, u, t)$ ($sim(u_j, u, t)$) is the similarity between u_i (u_j respectively) and u at time t by Eq.(3).

5 Experiments and Discussions

5.1 Experimental Sets

The experimental dataset WSDREAM records the invoking values of the real-world web service performance. WSDREAM contains 4 plain-text files, which are the users' information, the services information, the response-time values of the service invoking and the throughput values of the service invoking. For simplicity, we only use the response-time as the reliable values to predict.

In the following experiments, we compare FWSE-TCF with some state-of-the art approaches: UPCC[10](User-based collaborative filtering using Pearson Correlation Coefficient) predicts the evaluations only by the similar users; SPCC[11] (Service-based collaborative filtering using Pearson Correlation Coefficient) predicts the evaluations only by the similar services; TPCC[9](Time-aware collaborative filtering method using Pearson Correlation Coefficient) involves the non-linear reduction of the time-aware effectiveness into the process of the neighbor users finding.

To evaluate the performance of the various prediction approaches in reality, we randomly get rid of some data from the dataset and compare the differences between the predicted values and the original values. We use the sparsity to identify the numbers of the missing values. For example, 10% sparsity means that we get rid of 10% data and use the remaining 90% data to predict. The performance metrics are Mean Absolute Error(MAE) and Root Mean Squared Error(RMSE), which are defined as:

$$MAE = \sum\nolimits_{u_i,w_j} |h_{u_i,w_j} - h^*_{u_i,w_j}|/N \tag{5}$$

$$RMSE = \sqrt{\sum\nolimits_{u_i,w_j} (h_{u_i,w_j} - h^*_{u_i,w_j})^2/N} \tag{6}$$

Where h_{u_i,w_j} is the original value of the service w_j invoked by the user u_i, $h^*_{i,j}$ is the predicted value of the service w_j invoked by the user u_i with a approach, and N is the number of the prediction. Larger MAE (or RMSE) values indicate the worse prediction accuracy.

5.2 Studies on Parameters

Period T of Time-aware Quantification. The parameter T controls the decay period of the time-aware quantification. If T is smaller, the time-effectiveness would be the same for the most past data. While T is larger, the quantification would not guarantee the effectiveness. To achieve the suitable setting of the parameter T, we study the impacts of the parameter T by varying the values of T from 1 to 10 with a step value of 1. The values of other parameters are $n = 0.8$, $N = 10$ of Top-N, $f_0 = 1$ and $f_T = 0.01$. Fig.4 shows the MAE and RMSE results.

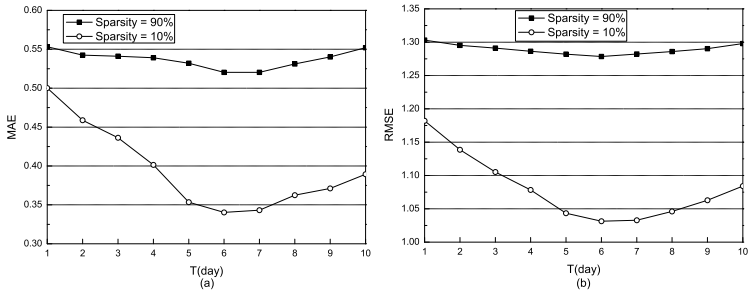


Fig. 4. Impact of T to MAE and RMSE

The experimental results of Fig.4 show that: (1) When T increases from 1 to 10, the prediction error will be improved; (2) When the sparisity = 90%, the fluctuation of MAE and RMSE are steady with the increase of T. While the sparsity is 10%, the fluctuation is obvious with the increase of T. (3) On the datasets with the sparsity of 90% and 10%, MAE and RMSE increase when decreasing T from 6 to 1, or increasing T from 6 to 10. Therefore, we set $T = 6$.

Exponent n of Time-aware Quantification. The parameter n controls the decay characteristics of the time-aware quantification. To get the suitable setting of the parameter n, we study the impacts of parameter n by varying the values of n from 1.5 to 0.5 with a step value of 0.2. The values of other parameters are $T = 6$, $N = 10$ of Top-N, $f_0 = 1$ and $f_T = 0.01$. Fig.5 shows the MAE and RMSE results.

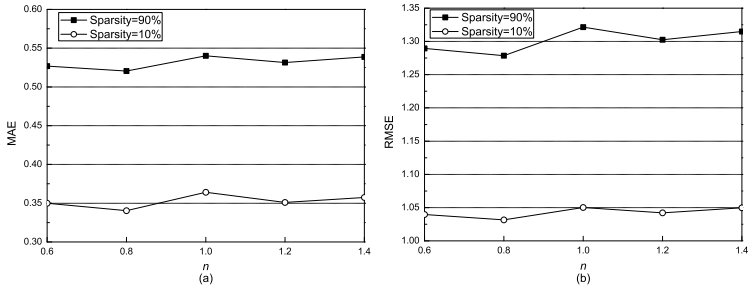


Fig. 5. Impact of n to MAE and RMSE

The experimental results of Fig.5 show that: (1) Compared with T, the sparsity have the similar impacts on the prediction error expect for the slight fluctuation of the prediction error with the increase of n; (2)For $n = 1$, the reduction process of the time-aware effectiveness becomes the linear reduction and the prediction error arrives maximum. When $n = 0.8$, the time-aware quantification would simulate the realistic the time-aware effectiveness of the past data. We set $n = 0.8$.

N **of Top-N.** The parameter N controls the numbers of the neighbor users. To get the suitable setting of the parameter N, we study the impacts of the parameter N by varying the values of N from 2 to 18 with a step value of 2. The values of other parameters are $T = 6$, $n = 0.8$, $f_0 = 1$ and $f_T = 0.01$. Fig.6 shows the MAE and RMSE results respectively.

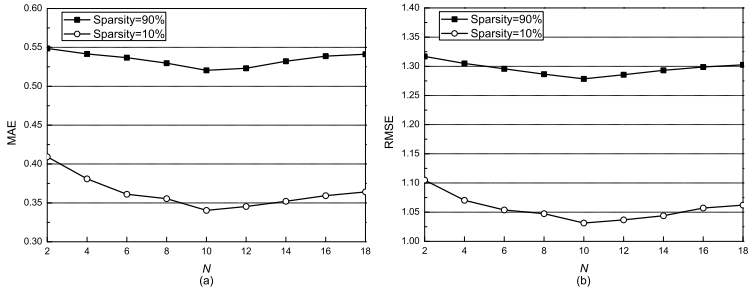


Fig. 6. Impact of N to MAE and RMSE

The experimental results of Fig.6 show that: FWSE-TCF achieves smallest prediction error when $N = 10$. On the datasets with the sparsity of 10% and 90%, MAE and RMSE decrease when increasing N from 2 to 10, or increasing when we increase N from 10 to 18. That is because when N is smaller than 10, it is not enough neighbor users for the prediction. When we increase N, the incremental users help to reduce the prediction error. However, while $N > 10$, the noise data are easy to involve in the prediction, and increase the prediction error slightly. Therefore, we set $N = 10$.

5.3 Comparison of Prediction Accuracy

Table 1 shows the results of MAE and RMSE of the different approaches on the datasets with the sparsity of 90%, 80%, 20% and 10%. The values of the parameters are $T = 6$, $n = 0.8$, $N = 10$ of Top-N, $f_0 = 1$ and $f_T = 0.01$.

Table 1. Performance comparisons of the different evaluation approaches

Sparsity	Metrics	SPCC	UPCC	TPCC	FWSE-TCF
90%	MAE	0.7596	0.5655	0.5424	0.5204
	RMSE	1.6133	1.3326	1.3018	1.2803
80%	MAE	0.7424	0.5516	0.5051	0.5041
	RMSE	1.6107	1.3114	1.2148	1.1850
20%	MAE	0.6703	0.4442	0.3769	0.3678
	RMSE	1.4102	1.1514	1.0622	1.0521
10%	MAE	0.6687	0.4331	0.3503	0.3432
	RMSE	1.4073	1.1264	1.0576	1.0328

Table 1 shows that: (1) The prediction error of the different approaches decrease with the decrease of the sparsity of the datasets. That is because the lower sparsity would provide enough data to predict; (2) Time-aware approaches (TPCC and FWSE-TCF) outperform the approaches without the time-aware effectiveness. That is because SPCC and UPCC ignore the time-aware effectiveness of the past data and use the outdated information to predict. According to the above analysis, the performance of the comparative approaches and proposed are SPCC<UPCC<TPCC<FWSE-TCF.

6 Conclusion and Future Work

In this paper we propose a time-aware reliability prediction approach for the web service evaluation. The main contribution is to quantify the time-aware effectiveness of the past data, which makes the recent data present more contribution than the outdated data in the process of the neighbor users finding. The comprehensive experimental analysis shows the accuracy of our proposed approach compared with the state-of-art approaches.

The proposed approach is applied in the component-based web service evaluation in this paper, while the system-based web service evaluation is the other important direction. Our on-going research includes applying the time-aware approaches into the system-based web service evaluation and design the evaluation aggregation mechanisms among the components.

Acknowledgement. This work is sponsored by the National Natural Science Foundation of China under Grant No. 61100007 and No. 61272186, the Provincial Natural Science Foundation under Grant No.F200937, the Basic Scientific Research Foundation of Harbin Engineering University under Grant NO.

HEUCF100608, No. HEUCF100613 and No. HEUCFZ1219, and the Provincial Foundation of Postdoctoral Sustentation under Grant No. LBH-Z12068.

References

1. Zibin, Z., Michael, R.: Lyu: Collaborative Reliability Prediction for Service-Oriented Systems. In: ACM/IEEE 32nd International Conference on Software Engineering, pp. 35–44. ACM Press, South Africa (2010)
2. MovieLens Data Sets, http://www.grouplens.org/node/73/
3. Rong, W., Liu, K., Liang, L.: Personalized Web Service Ranking via User Group Combining Association Rule. In: Anon. 2009 IEEE International Conference on Web Services, pp. 445–452. IEEE Press, Los Angeles (2009)
4. Zibin, Z., Yilei, Z., Lyu, M.R.: Distributed QoS Evaluation for Real-World Web Services. In: 8th International Conference on Web Services, p. 8. IEEE Press, Miami (2010)
5. Web Service QoS Datasets, http://www.wsdream.net/dataset.html
6. Junsheng, C., Huaimin, W., Gang, Y.: DyTrust:A Time-Frame Based Dy-namic Trust Model for P2P Systems. Chinese Journal of Computer 29, 1301–1307 (2006) (in Chinese)
7. Xiaoyong, L., Xiaolin, G.: Trust Quantitative Model with Multiple Decision Factors in Trusted Network. Chinese Journal of Computer 32, 405–416 (2009) (in Chinese)
8. Koychev, I., Schwab, I.: Adaptation to Drifting User's Interests. In: ECML2000 Workshop: Machine Learning in New Information Age, pp. 39–46. IEEE Press, United States (2000)
9. Hong, Y., Zhuanyun, L.: A Collaborative Filtering Recommendation Algorithm based on Forgetting Curve. Journal of Nanjing University(Natural Sciences) 46, 520–527 (2010) (in Chinese)
10. Shao, L., Zhang, J., Wei, Y., et al.: Personalized QoS Prediction for Web Services via Collaborative Filtering. In: IEEE International Conference on Web Services, pp. 439–446. IEEE Press, Salt Lake City (2007)
11. Sarwar, B., Karypis, G., Konstan, J., et al.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th international conference on World Wide Web, pp. 285–295. ACM Press, New York (2001)

An Improved PBIL Algorithm for Path Planning Problem of Mobile Robots

Qingbin Zhang [1,2], Manjun Cai[3], Fajun Zhou[1], and Hairong Nie[1]

[1] Center of Green & Intelligent Engineering,
Shijiazhuang Institute of Railway Technology, Shijiazhuang 050061, China
[2] Key Laboratory of Advanced Design and Intelligent Computing (Dalian University),
Ministry of Education, Dalian 116622, China
[3] Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
zqbin2002@sina.com

Abstract. The path planning problem of mobile robots is a NP-Hard problem often solved by evolutionary approaches such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO). However, the algorithm's performance is often influenced heavily by the determination of the operators and the choice of related parameters. In this paper, a permutation code PBIL is proposed to solve the path planning problem. First, a free space model of the mobile robot is constructed by the MAKLINK graph; second, a sub-optimal path is generated by the Dijkstra algorithm; then global optimal path is constructed by the permutation code PBIL based on the sub-optimal path. Simulation results show that the PBIL can get satisfied solutions more simply and efficiently with fewer operators and parameters.

Keywords: Path planning problem, MAKLINK graph, Permutation code PBIL.

1 Introduction

The path planning problem of autonomous mobile robots is to find a collision-free path from a starting point to a goal in a given environment according to distance, time or cost [1]. Research results show that the classic approaches, such as the potential field method [2], visibility graph methods [3], and grid methods [4] tend to get local optimal solution which may be far inferior to the global optimal solution, so it is more appropriate to adopt heuristic based evolutionary approaches to solve the problems [5]. Among those approaches, Genetic Algorithm (GA) [6] and Ant Colony Optimization (ACO) [7] have been widely employed to find the globally optimal path.

Sun Shudong and Lin Mao [8] presented a strategy for path planning of multi-mobile robots using GA. MAKLINK graph was used to model the workspace of robots. Sequence numbers was used to mark the cross points on the MAKLINK graph and also as an encoding method for the chromosomes of GA. Simulation results show that the method is effective in path planning of multi-mobile robots. Nagib and Gharieb[9] constructed fixed-length path using GA consisting of binary strings in a grid model. The proposed algorithm could find a quick solution for environments with few

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 85–92, 2013.

obstacles. Al-Taharwa et al.[10] adopted GA with an integer string encoding method to solve the mobile robot path planning problem in a grid environment model. Experiments show that the proposed approach is effective and efficient in handling different types of tasks in static environments.

Tan Guanzheng [11] proposed the ACS to find the globally optimal path of mobile robots based on MAKLINK graph model and the initial sub-optimal collision-free path constructed by Dijkstra algorithm. Results show that ACS performed better than that of GA. Jing Zhou et al. [12] proposed two algorithms based on ACO framework, they applied visibility graph as the roadmap and construction graph. Experimental results illustrated that the proposed algorithms are effective and a flexible balance between the optimum and the number of iterations could also be achieved.

However, the research has also shown that the performance of GA and ACS for the path planning problems of mobile robots depends heavily on determination of the operators as well as choice of related parameters, which needs prior knowledge about the problem at hand, otherwise extensive experiments have to been done to choose an effective configuration of the operators and the parameters [13].

The PBIL is one of the simple optimization algorithm belongs to the Estimation of Distribution Algorithms (EDAs) paradigm [14, 15]. PBIL applies a probability vector to represent the population, with each element referring to the probability of obtaining a given value in a specific gene position. The probability vector is updated towards the selected solutions by Hebbian rule. PBIL has been proved to be more successful on many benchmark functions and has been used in some real-world problems [16].

The initial PBIL uses a binary code string to represent the variables. In this paper, a permutation code PBIL to solve the path planning problem is proposed. Experimental results compared with that of the ACS are reported.

2 The Path Planning Problem

In order to solve the global optimal path efficiently, the path planning problem is often divided into two subtasks: 1) establish the environment model of the mobile robot. Various types of known static environment models have been proposed by previous robot path planning researchers [13], such as Visibility graph[3], MAKLINK graph[11], and Cell Decomposition[10] are used to produce the connectivity of graph for robot path planning. 2) Searching the graph for a sequence of vertices which connect the start and goal position. After the graph model has been constructed, the path planning is thereby converted into a problem searching for an optimal path between the initial and the final of the points in the given graph. Usually, the Dijkstra algorithm or the A* algorithm is used [17].

2.1 Environment Model of the Mobile Robot

MAKLINK graph and grid model are two mainly tools to model the space of mobile robots. In this section, a free space model of the mobile robot is constructed by the MAKLINK graph theory. The free space in an environment means the space in which the robot can move freely. To establish the free space model, the following assump-

tions need to be made: 1) the heights of the environment and obstacles can be ignored; 2) both the environment and the obstacles have a polygonal shape; 3) in order to avoid a moving path is too close to the obstacles, the boundaries of every obstacle can be expanded and then the size of the robot can be ignored.

Assume the total number of the free MAKLINK lines on a MAKLINK graph is l, the middle points of these free MAKLINK lines can be denoted by v_1, v_2,…, v_l, respectively. If we denote the starting point S and goal T as v_0 and v_{l+1}. Then points set V={ v_0, v_1,…, v_{l+1} } is the set including points S and T and the middle points of all the free MAKLINK lines. Further more, set E is defined as a set of the lines which includes: the lines connecting point S and the middle points on the free MAKLINK lines adjacent to S; the lines connecting a pair of the middle points on two adjacent free MAKLINK lines; and the lines connecting the goal T and the middle points on the free MAKLINK lines adjacent to T. thus the undirected graph G (V, E) can be respected as the free space model, the path planning can be converted into finding the shortest path between the given starting point S and goal T on G (V, E).

2.2 Searching for Sub-optimal Path Using Dijkstra Algorithm

In order to find the shortest path between the starting point S and goal T by Dijkstra algorithm, it is necessary to define the adjacency matrix with weights for the network graph G (V, E) in which each element of the matrix represents the length of the straight line between two adjacent path nodes on G (V, E), where a path node means the intersection of the robot path and a free MAKLINK line. Each element of the matrix is defined as follows:

$$adjlist[i][j] = \begin{cases} length(v_i, v_j), & if\ edge(v_i, v_j) \in E \\ \infty, & others \end{cases} \tag{1}$$

Where $adjlist[i][j]$ is the element corresponding to the i^{th} row and the j^{th} column of the matrix, $length(v_i, v_j)$ is the straight-line distance between the path nodes v_i, v_j, i and j=0, 1, 2,…, l, l +1.

The shortest path generated by the Dijkstra algorithm is just a sub-optimal path because all the path nodes are middle points of the relevant free MAKLINK lines, but it can be adopted as the initial path for evolutionary approaches to search the globally optimal path.

2.3 Description of the Optimization Problem

Assume that the sub-optimal path is denoted in order by path nodes P_0, P_1,, …, P_d, and P_{d+1}, where P_0 and P_{d+1}denote the starting point S and the goal T, respectively. We can adjust the position of each path node in the free MAKLINK lines to construct new paths and then optimize the location of the path nodes so as to generate the globally optimal path. For each point in a free MAKLINK line, its location can be expressed as:

$$P_i = P_{i1} + (P_{i2} - P_{i1}) \times h_i, \quad h_i \in [0,1] \tag{2}$$

So different combinations of parameters h_1, h_2,…, h_d will generate different paths. Thus finding the shortest path on MAKLINK graph can be regarded as another optimization problem, which is to find the optimal parameter set { h_1, h_2,…,h_d }such that the following objective function has the minimum value.

$$L = \sum_{i=0}^{d} length\{P_i(h_i), P_{i+1}(h_{i+1})\} \tag{3}$$

Where length (P_i, P_{i+1}) represents the straight line distance between two adjacent path nodes P_i, and P_{i+1} on the lines L_i and L_{i+1}.

In order to solve the new optimization problem, we divide equally each of the MAKLINK lines into D portions, and thus (D+1) nodes are generated on each line, which represent the (D+1) possible values of parameter h_i (i=0,1, 2,…, d). Then the solution space is $d \times (D+1)$ nodes in total. For example, if we divide equally each of the MAKLINK lines into ten portions, and thus eleven nodes on each line represented as h_i are 0, 0.1, 0.2…, and1.0.

3 PBIL for Optimal Path Planning Problem

3.1 PBIL Algorithm

The initial PBIL uses a binary code in which the population is represented by a probability vector:

$$p_l(x) = (p_l(x_1), p_l(x_2), ...p_l(x_n))$$

Where $p_l(x_i)$ refers to the probability of obtain a value 1 in the i^{th} gene position in the l^{th} generation. The algorithm begins by initializing the probability vector $p_0(x)$ with all elements set to 0.5, which means random solutions are created when sampling from this probability vector. As search progresses, the values in the probability vector are gradually learned towards values representing high fitness solutions.

The evolution process in PBIL is accomplished as follows: At each generation, M solutions are generated based upon the probabilities specified in the current probability vector $p_l(x)$. The solutions are then evaluated and N best solutions ($N \le M$) are selected as the best solutions set. We denote them by $x^l_{k:M}$ (k=1,…, N). These selected best solutions are used to update the probability vector by using a Hebbian inspired rule:

$$p_{l+1}(x) = (1-a)\,p_l(x) + a\frac{1}{N}\sum_{k=1}^{N} x^l_{k:M} \tag{4}$$

Where a is the learning rate.

After the probability vector is updated, a new set of solutions are generated by sampling from the new probability vector $p_{l+1}(x)$, and the cycle is repeated until some

termination condition is satisfied, e.g., the probability vector is converged to either 0.0 or 1.0 for each bit position, or the number of iterations is met.

3.2 Permutation Code PBIL

In the permutation code PBIL, the probability vector represented the direction of the evolution in binary PBIL is expanded to be a probability matrix P. Suppose the solution of a permutation optimization problem is represented by an integer string, the length of the integer string is n and the value range is from 1 to m, then an element p_{ij} in P represents the probability which the value is j in the i^{th} gene ($1 \le i \le n$; $1 \le j \le m$).

The initial value of the probability matrix is a uniform distribution. During the evolution, the values of the p_{ij} is updated by the formula (4) and roulette method is used to sample from the probability matrix in order to get the solutions of the problem.

3.3 PBIL for Optimal Path Planning Problem

The PBIL for path planning problem of a mobile robot between a given starting point S and a given goal T in a given environment can be described as follows.

Step1. Construct the free space model of the robot using the MAKLINK graph, and then find a sub-optimal collision-free path in the free space using the Dijkstra algorithm;

Step2. Define the initial probability matrix P with each element p_{ij} as $1/(D+1)$, Sample from the probability matrix P to form the initial solutions set S_1 with M paths;

Step3. Set the iteration counter $t=1$, Repeat for $t=1,2,\ldots$until the number of iterations NC is met;

Step4.Select $N(N<M)$ individuals from S_t according to the length of different paths, the path with the minimum length is represented as $B(t)$;

Step5.Update the probability matrix P with formula (4);

Step6. Sample from the probability matrix P with roulette wheel selection method to generate M new paths to form the new set S_{t+1};

Step7. Set $t=t+1$, if $t < NC$ return to Step 4. Else, output the terminal probability matrix P, and the best solution set B.

4 Experiments and Results

4.1 The Example Problem

The example moving environment of a mobile robot is a 200 by 200 meters square and includes four obstacles. The coordinates of vertices of the four obstacles are(40 140;60 160;100 140; 60 120),(50 30;30 40; 80 80; 100 40), (120 160;140 100;180 170;165 180),(120 40;170 40;140 80),respectively. The(x,y)coordinates of the starting point S and the goal T are(20,180) and (160,90), respectively[18].

For the example problem, the free space graph of the robot is shown in Fig.1, where number of the free MAKLINK lines $l=20$. The sub-optimal path is obtained by

Dijkstra algorithm is *S*—v8—v7—v6—v12—v13—v11—*T*, which is shown in Fig. 2.The length of this path is180.78 meters.

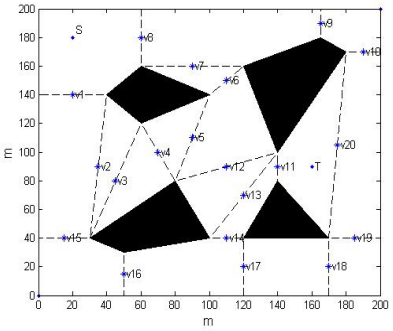


Fig. 1. The free space of the robot

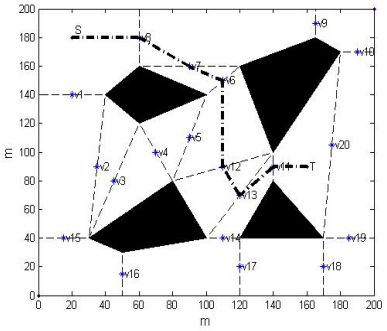


Fig. 2. The optimal path obtained by Dijkstra algorithm

4.2 Experimental Results

The solution of the PBIL is a probability matrix P in which each element p_{ij} means in the i^{th} line L_i, the probability of selects the j^{th} point. i=(1,2,…d), j=(1,2,…D, D+1). However, we can construct the optimal path by sampling from the matrix.

To demonstrate the performance of the proposed PBIL on the globally optimal path planning problem, parameters are set as follows: the sample paths in each iteration M =200; in which number of selected as the best paths set N=0.2*M; the learning rate a is set to be 0.05, D=10. Experimental results of the average value of probability matrix P based on 100 separate running with 500 iterations are presented in Table1. From the Table 1, we can conclude that the algorithm will evolve towards some positions with high probability on each line of the path and the probability of other points will converge to zero. So we can construct different paths following different demands.

Table 1. Average value of probability matrix P

No. of the lines	Probability of each position on different lines										
1	0.12	0.86	0.02	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00
2	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0.21	0.67	0.11
3	0.96	0.04	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00
4	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0.55	0.45
5	0.54	0.46	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00
6	0.39	0.06	0. 00	0.47	0.08	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00

A typical running comparing of the PBIL and ACS for the path planning problem of mobile robots is shown in Fig.3. The path founded by PBIL and that of ACS is shown in Figure 4. the operators and parameters of ACS are set same as in [7].The length of the path founded by PBIL is 168.67 meters, however, the length of the path founded by ACS is173.82 meters, so compared with the ACS, PBIL can find shorter path with faster converge speed.

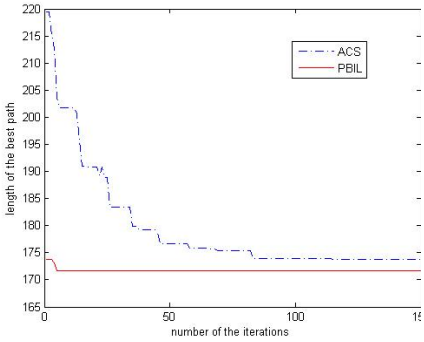


Fig. 3. Typical running of the PBIL and ACS

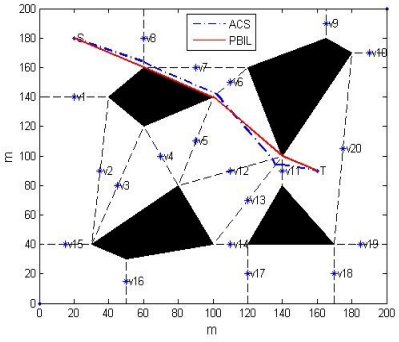


Fig. 4. The optimal path obtained by PBIL and ACS

5 Conclusion

In this paper, a permutation code PBIL is proposed to solve the path planning problem of autonomous mobile robots. The free space model of the mobile robot is constructed by the MAKLINK graph and the initial sub-optimal path is generated by the Dijkstra algorithm. The global optimal path is constructed by sampling from the probability matrix of the permutation code PBIL. Simulation results compared with that of the ACS on experimental problem show that proposed PBIL can solve the problem more simply and efficiently with less operators and parameters.

Acknowledgements . This work was supported by Key Laboratory of advanced Design and Intelligent Computing (Dalian University), Ministry of Education (ADIC2010005).

Refences

1. Raja, P., Pugazhenthi, S.: Optimal path planning of mobile robots: A review. International Journal of Physical Sciences 7(9), 1314–1320 (2012)
2. Arambula, F., Padilla, M.: Autonomous Robot Navigation Using Adaptive Potential Fields. Mathematical and Computer Modeling 40, 1141–1156 (2004)

3. Roh, S.G., Park, K.H., Yang, K.W.: Development of Dynamically Reconfigurable Personal Robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4023–4028 (2004)
4. Payton, D.W., Rosenblatt, J.K., Keirsey, D.: Grid-based mapping for autonomous mobile robot. Robot. Auton. Syst. 11(1), 13–21 (1993)
5. Hu, Y., Yang, S.X.: A Knowledge Based Genetic Algorithm for Path Planning of Mobile Robot. In: IEEE Int. Conf. on Robotics and Automation, vol. 5, pp. 4350–4355 (2004)
6. Goldberg, D.E.: Genetic Algorithms in Search,Optimization and Machine Learning. Addison-Wesley Longman publishing Co., Inc., Boston (1989)
7. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press/Bradford Books, Cambridge (2004)
8. Sun, S., Lin, M.: Path Planning of multi-mobile robots using genetic algorithms. Acta Automatica Sinica 5(26), 672–676 (2000)
9. Nagib, G., Gharieb, W.: Path planning for a mobile robot using genetic algorithms. In: Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEEC 2004), pp. 185–189 (2004)
10. AL-Taharwa, I., Sheta, A., Al-Weshah, M.: A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment. Journal of Computer Science 4(4), 341–344 (2008)
11. Tan, G., He, H.: Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots. Acta Automatica Sinica 33(3), 279–285 (2007)
12. Zhou, J., Dai, G., He, D.Q., Ma, J., Cai, X.-Y.: Swarm Intelligence: Ant-based Robot Path Planning. In: Fifth International Conference on Information Assurance and Security, pp. 459–463 (2009)
13. Buniyamin, N., Sariff, N., Wan Ngah, W.A.J., Mohamad, Z.: Robot global path planning overview and a variation of ant colony system algorithm. International Journal of Mathematics and Computers in Simulation 5(1), 9–16 (2011)
14. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Carnegie Mellon University, Technical Report CMU-CS-94-163 (1994)
15. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
16. Wang, Z., Zhang, Q., Ma, Y., Zhang, J., Liu, Y.: An Improved PBIL Algorithm for the Machine-Part Cell Formation. Applied Mechanics and Materials(26 - 28), 498–501 (2010)
17. Sariff, N., Buniyamin, N.: An overview of autonomous mobile robot path planning algorithms. In: 4th Student Conference on Research and Development, pp. 183–188 (2006)
18. Li, M.: Modeling and Path Planning of Mobile Robot. Yanshan University (2012)

An Initialized ACO for the VRPTW

Wei Shi and Thomas Weise

The USTC-Birmingham Joint Research Institute
in Intelligent Computation and Its Applications (UBRI)
University of Science and Technology of China (USTC)
Hefei, Anhui, China

Abstract. The Vehicle Routing Problem with Time Windows is an important task in logistic planning. The expenditure on employing labor force, i.e., drivers for vehicles, accounts for most of the costs in this domain. We propose an initialized Ant Colony approach, *IACO-VRPTW*, with the primary goal (f_1) to reduce the number of vehicle needed to serve the customers and the second-priority goal (f_2) of decreasing the travel distance. Compared with methods that optimize f_2, *IACO-VRPTW* can reach or reduce f_1 in 8 out of 18 instances of the Solomon benchmark set, at the cost of increasing travel distance slightly. *IACO-VRPTW* can effectively decrease the number of vehicles, travel distance and runtime compared with an ACO without initialization.

1 Introduction

The Vehicle Routing Problem with Time Windows (VRPTW) is a common and important task in logistic planning. Here, the goal is to use capacity-restricted vehicles to serve several customers that require certain amounts of a product. Most related works try to minimize either first the number f_1 of these vehicles and then the distance these vehicles travel (f_2), or only focus on f_2.

In this paper, we present the results of a new Ant Colony Optimization (ACO) [1] approach – *IACO-VRPTW* – for solving VRPTWs. *IACO-VRPTW* can be distinguished from the related works by three main differences:

1. It does not follow a two-step approach, but optimizes both goals f_1 and f_2 all the time during its run (but *not* in a Pareto fashion).
2. It gives f_1 strict priority over f_2, i.e., a solution which requires fewer vehicles is better and amongst solutions with the same f_1-values, the one with the shortest travel distance f_2 is best. This approach is closer to practice, as costs for drivers are usually higher than costs for travel distance.
3. It applies an initialization procedure of the transition matrix in order to achieve better results earlier.

We will first outline the VRPTW in the next section and then discuss related works in Section 3. In Section 4 we describe our new approach. Experimental results are given in Section 5 and in Section 6, we conclude our paper and list our plans for future work.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 93–100, 2013.

2 Problem Definition

The Vehicle Routing Problems with Time Windows (VRPTW) is an extension of the VRP. Given are a set V of $m = |V|$ vehicles v_i, each having the same fixed capacity $k \in \mathbb{N}$ and being parked at the central depot c_0. Additionally, there is a set C with $n = |C|$ cities c_i to be serviced ($i \in 1..n$).

Each city c_i has a demand for a certain amount $w_i \in \mathbb{N}$ of a product and must be serviced by exactly one vehicle. The delivery of the product must begin between the earliest arrival time $e_i \in \mathbb{N}$ and latest arrival time $l_i \in \mathbb{N}$ of that city and takes the service time $s_i \in \mathbb{N}$. Vehicles cannot leave the depot before e_0 or arrive back at the depot after l_0.

The travelling cost is the distance d_{ij} between two cities c_i and c_j. In the available benchmark cases, this usually is the Euclidean distance and city locations are points in the two-dimensional plane. The travel time is the distance divided by *speed*. In benchmark cases usually *speed* = 1.

A solution $r = (r_1, r_2, \ldots, r_m)$ of a VRPTW describes which city should be serviced by which vehicle. For each vehicle v_i with $i \in 1..m$, it provides an *ordered* set $r_i \subseteq 1..n$ describing the sequence of cities to visit. It is implied that vehicles start at the depot and return to it after finishing their schedule.

$$n = \sum_{i=1}^{m} |r_i| \tag{1}$$

$$\forall i, j : 1 \leq i \neq j \leq m \Rightarrow r_i \cap r_j = \emptyset \tag{2}$$

$$\text{feasible solution} \Rightarrow \forall i : 1 \leq i \leq m \Rightarrow k \geq \sum_{\forall c \in r_i} w_c \tag{3}$$

For any valid solution r, a set of conditions must hold, i.e., all customers must be serviced (Eq. 1), be serviced only once (Eq. $1 \wedge 2$), and no vehicle's capacity must be exceeded (Eq. 3).

Each vehicle v_i processes its tour r_i step by step from the beginning. The earliest service time $b_{r_{i,j}}$ for city $r_{i,j}$, i.e., the j^{th} city in schedule r_i (for vehicle v_i) is then given as Eq. 4. It is determined by the earliest arrival time $e_{r_{i,j}}$ of that city and the earliest service time $b_{r_{i,j-1}}$ of the city $r_{i,j-1}$ serviced before it, or the earliest departure time of the depot c_0, if it is the first city in schedule r_i ($j = 1$).

$$b_{r_{i,j}} = \begin{cases} \max\{e_{r_{i,j}},\ e_0 + t_{0\,r_{i,j}}\} & \text{if } j = 1 \\ \max\{e_{r_{i,j}},\ b_{r_{i,j-1}} + s_{r_{i,j-1}} + t_{r_{i,j-1}\,r_{i,j}}\} & \text{if } j > 1 \end{cases} \tag{4}$$

$$\forall i : 1 \leq i \leq n \Rightarrow e_i \leq b_i \leq l_i \tag{5}$$

Of course, in a feasible solution, all cities can be serviced within their respective time windows, as shown in Eq. 5.

Solving VRPTW generally has three goals: *(a)* use as few vehicles as possible (f_1, Eq. 6), *(b)* aim for the lowest travel cost (f_2, Eq. 7), and *(c)* minimize the vehicles' total waiting time. Yu et al. [2], Alvarenga et al. [3] choose the total travel distance f_2 as their objective. Homberger and Gehring [4] combined the

number of vehicles f_1 and f_2 as weighted sum in the objective function, while Li and Lim [5] include all three objectives as well as the schedule time. Common are also two-step processes [6, 7, 8], as we will outline in Section 3.

In the real world, using fewer vehicles will greatly decrease the transportation cost compared with more vehicles but slightly shorter distance. Thus, when we compare two solutions a and b, the one with fewer tours f_1 is considered as better. If $f_1(a) = f_1(b)$, then the one with the lower cost f_2 will win.

$$f_1(r) = |\{\forall i \in 1..m : |r_i| > 0\}| \tag{6}$$

$$f_2(r) = \sum_{i=1}^{m} \left[d_{0\, r_{i,1}} + d_{r_{i,|r_i|}\, 0} + \sum_{j=2}^{|r_i|} d_{r_{i,j-1}\, r_{i,j}} \right] \tag{7}$$

3 Related Work

Solomon [9] gives a detailed description of the VRPTW and the benchmark set used in most of the related works (and here as well). This set consists of six subsets: R1, C1, R2, C2, RC1, RC2. R2, C2, and RC2 are problems with a long scheduling horizon compared with R1, C1, and RC1, i.e., its trucks have a greater capacity and the time windows are broader.

As the VRPTW is NP-hard [9], meta-heuristics are used to approximately solve them. In order to satisfy the multiple objectives of VRPTW, Gambardella et al. [10] introduced a two-step approach, which then was used in many other works: Firstly they minimize the number of tours and secondly, under the given number of vehicles, minimize the total travel cost. Gambardella et al. [10] used the same metaheuristic for both steps. Berger and Barkaoui [6] adopted a Genetic Algorithm (GA) in both two phases. In the algorithm of Homberger and Gehring [8], the number of vehicles is minimized with an evolution strategy and the total distance by means of tabu search. Bent and Hentenryck [11] proposed a robust heuristic approach for VRPTW, based on simulated annealing and linear neighborhood spreads.

In our approach, both objectives are optimized at once. They are not combined to a weighted sum, but prioritized. We try to find schedules requiring fewer vehicles (f_1) and, at the same time, lower total travel cost (f_2).

We therefore use an ACO. The ACO idea has first been proposed in [1] and is inspired by the way how ants find paths based on pheromone released by other ants. In ACO, a solution is represented as pathes through a graph constructed by simulated ants. Like in nature, these ants start at a certain location and move forward, basing their decision of where to go on *(a)* pheromone τ (a dynamic value that can be changed) and *(b)* a sensed distance d to food (i.e., a static heuristic value). Ants that find good solutions may lay out additional pheromone $\Delta\tau$ that supports later ants to find short path.

Qi and Sun [7] proposed an ACO for VRPTW that optimizes f_1 and f_2 separately and outperforms the algorithm by Gambardella et al. [10]. Yu et al. [2] propose hybrid ACO algorithms that use local searches, such as tabu search

and neighborhood search for VRPTW. Instead of a local search, we use a special initialization procedure.

4 Approach: Initialized ACO

4.1 Preprocessing

The service beginning time b_i of c_i depends on the cities on the same tour visited before c_i. Thus, it is impossible to know the b_i of a city c_i until the cities visited before it have been chosen. However, we can rule out certain cities that would violate Eq. 5:

$$\text{can visit } c_j \text{ after } c_i \text{ in schedule } r_k \Rightarrow l_j \geq e_i + s_i + t_{i\,j} \tag{8}$$

Eq. 8 is one necessary condition for being able to reach c_j in time after servicing c_i. For each city c_i, we thus calculate the set of other cities that can be visited afterwards in the same schedule. We call this set the *domain* of c_i.

When searching for the next city to be visited, we just need to consider the cities in the current city's domain. This reduces the search space size and speeds up the optimization process.

4.2 ACO Strategy

In our algorithm, we use an elitist pheromone update strategy of type ANT-cycle [1]. Compared with ANT-quantity and ANT-density, ANT-cycle performs better as it updates pheromone globally not locally [1]. The updating rule for ANT-cycle is defined in Eq. 9, where $\Delta\tau$ denotes the pheromone change after one algorithm iteration:

$$\tau_{i\,j} = \rho\tau_{i\,j} + \Delta\tau_{i\,j} + \Delta\tau^{\star}_{i\,j} \tag{9}$$

$$\Delta\tau_{i\,j} = \sum_{k=1}^{a} \Delta\tau^{k}_{i\,j} \tag{10}$$

$$\Delta\tau^{k}_{i\,j} = \begin{cases} \frac{Q_1}{f_2^k} & \text{if ant } k \text{ travels from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\Delta\tau^{\star}_{i\,j} = \begin{cases} \frac{Q_2}{d_{ij}m^{\star}} & \text{if the elitist ant travels from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Here, ρ is the evaporation coefficient that reduces the impact of the previous pheromone, a is the number of ants, Q_1 is the total quantity of pheromone an ant can leave on the way from c_i to c_j and f_2^k is the total travel distance of ant k. Eq. 11 is the normal pheromone update definition of ANT-cycle. $\Delta\tau^{\star}_{i\,j}$ in Eq. 12 is an extra prize for the elitist ant, i.e., the one that used the fewest vehicles (if there exists multiple such ants, then it is the one with the shortest travel distance). We modified the original formula of the update [1] to consider

both, the number of vehicles required of the elitist ant and the travel distance in Eq. 12.

The $(n+1) \times (n+1)$ pheromone matrix τ is used to calculate the transition probability matrix P according to Eq. 13, where $p_{i\,j}$ is the probability of an ant transferring from c_i to c_j.

$$P = \{p_{ij}\}, \quad p_{ij} = \begin{cases} \frac{(\tau_{i\,j})^{\alpha} * (1/d_{i\,j})^{\beta}}{\sum_j (\tau_{i\,j})^{\alpha} * (1/d_{i\,j})^{\beta}} & \text{if } c_j \text{ unvisited} \wedge \text{in domain of } c_i \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The index α determines the relative influence of the pheromone trails and β is the influence of the visibility. The probability $p_{i\,j}$ increases with higher pheromone density $\tau_{i\,j}$ and shorter distance $d_{i\,j}$.

In our algorithm, we have a certain number a of ants at the depot. Each of them must visit all the cities under the capacity and time window constraints. When an ant reaches its maximum capacity or has no next city to visit, it will automatically return to the depot. In the former case, its capacity is replenished and it begins the next tour.

4.3 Initialization Procedure

We use a simple directional selection to initialize the transition probability matrix P for the first ACO iteration. When we start the search, we set the transition probability from the depot to be higher for those cities which are close and have an earlier earliest start time. The top m cities with the lowest $h_{0\,j}$-values (Eq. 14) thus receive higher probabilities $p_{0\,i}$. The values γ_1, γ_2, δ_1, and δ_2 are weights.

$$h_{0\,i} = \gamma_1 d_{0\,i} + \delta_1 e_i \tag{14}$$

$$h_{i\,j} = \gamma_2 d_{i\,j} + \delta_2 d_{j\,0} \tag{15}$$

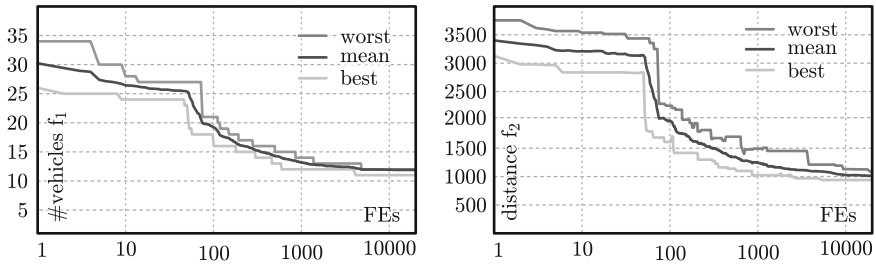
For ants leaving a city $i \neq 0$, we use the similar equation 15, which incorporates the distance back to the depot. The goal is to avoid that the ants end their tours in cities very far away from the depot. From the domain of each city c_i, we chose the top-num_1 cities according to Eq. 15. Amongst them, num_2 cities $num_2 < num_1$ are randomly chosen and receive higher probabilities.

5 Experiments

5.1 Experimental Setup

In our experiments, we use Solomon's benchmark for VRPTW [9]. Each complete solution r constructed by an ant will be counted as one function evaluation (FE). We grant 20 000 FEs to runs on 25 and 50 city problems and 300 000 FEs for 100 city problems. We use an Intel Core i3-3220 CPU with 3.30GHz.

The parameters of our algorithm have been introduced in Section 4. We set $Q_1 = 4000$, $Q_2 = 80$, $\rho = 0.9$, $num_2 = 8$, $\gamma_1 = 1$, $\delta_1 = 1$, $\gamma_2 = 1.5$ and



(a) Progress in terms of f_1. (b) Progress in terms of f_2.

Fig. 1. Convergence diagrams over all runs

$\delta_2 = 0.5$. Then we test all combinations of $a \in \{10, 20, 30, 40, 50\}$, $\alpha \in \{0.5, 1\}$, and $\beta \in \{12, 3, 4\}$. For the 25-city problem, we set $num_1 \in \{10, 20\}$ and for the 50 and 100-city problems, we use $num_1 \in \{15, 25, 45\}$. We run the experiments with and without the initialization procedure discussed in Section 4.3, for 30 runs per setup.

5.2 Experimental Results

We compare the arithmetic means of the solution data over all configurations and find that $a = 50$, $\beta = 2$, and $\alpha = 1$ together with $num_1 = 20$, $num_1 = 25$, and $num_1 = 45$ for the 25, 50, and 100 city problems, respectively, perform best, i.e., lead to lower f_1 and f_2 values.

In Fig. 1, we illustrate the progress of our algorithm with that configuration for 100-city problem C101. The algorithm behaves similarly on the other instances. While f_1 decreases steadily, f_2 sometimes increases. This happens when new solutions with fewer vehicles appear that initially have a longer travel distance.

We now compare our results obtained with the above settings with the solutions from [2, 12] that were obtained by highly specialized algorithms. From Table 1, we see that the mean number $\overline{f_1}$ of vehicles of 8 out of 18 problem types are equal to or smaller than in [2, 12]. As our primary goal is to find solutions with fewer vehicles and then minimize the total travel distance, the results show that *IACO-VRPTW* is effective. It performs particularly well in problems with longer scheduling horizon (C2, RC, and RC2). Our method has some problems with the larger-scale instances. It can be seen that *IACO-VRPTW* finds good solutions in short time (less than 150s) for 100-city problems. This is probably due to the very limited number of FEs granted to the experiments, but we needed that setting for these first experiments to identify good configurations.

In Table 2, we compare our algorithm (*IACO-VRPTW*) to an algorithm version without initialization (*ACO-VRPTW*). The results of type C1, R1, and RC1 with 25 cities are aggregated under point 25_1, those of C2, R2, and RC2 under 25_2, and so on. From the table, it is obvious that the initialization procedure improves the result quality significantly and even the runtime.

Table 1. Comparision of *IACO-VRPTW* with the results published in [2, 12] in terms of mean objective values ($\overline{f_1}$, $\overline{f_2}$) and mean runtime $\overline{\text{RT}}$

Instance	From [2, 12] $\overline{f_1}$	From [2, 12] $\overline{f_2}$	*IACO-VRPTW* $\overline{f_1}$	*IACO-VRPTW* $\overline{f_2}$	relative error ϵ ϵf_1	relative error ϵ ϵf_2	Runtime $\overline{\text{RT}}$ in s
25_C1	3.00	190.59	3.00	195.27	**0.00%**	2.46%	0.87
25_R1	4.92	462.87	4.92	490.15	**0.00%**	5.89%	0.96
25_RC1	3.25	350.24	3.50	374.01	7.69%	6.79%	0.95
25_C2	2.00	214.46	1.63	221.34	**-18.75%**	3.11%	0.81
25_R2	2.73	382.17	1.64	434.12	**-39.92%**	13.59%	0.91
25_RC2	2.88	319.53	1.88	370.99	**-34.78%**	16.11%	0.86
50_C1	5.00	361.69	5.22	400.22	4.40%	10.66%	2.45
50_R1	7.75	766.13	8.25	878.89	6.45%	14.71%	3.60
50_RC1	6.50	729.24	7.38	837.84	13.46%	14.89%	2.63
50_C2	2.75	357.50	2.38	397.71	**-13.64%**	11.25%	2.32
50_R2	4.11	634.03	2.50	782.35	**-39.19%**	23.39%	2.64
50_RC2	4.29	585.24	3.71	841.90	**-13.33%**	43.86%	2.73
100_C1	10.00	826.70	10.78	1074.38	7.78%	29.96%	115.61
100_R1	12.75	1155.89	16	1506.22	25.49%	30.31%	130.39
100_RC1	12.38	1342.42	15.63	1719.98	26.26%	28.13%	122.35
100_C2	3.00	587.36	3.88	788.92	29.17%	34.31%	143.55
100_R2	3.73	906.28	4.45	1331.70	19.51%	46.94%	144.89
100_RC2	4.25	1049.57	5.38	1613.93	26.47%	53.77%	126.79

Table 2. Comparison of our method with initialization (*IACO-VRPTW*) to the approach without (*ACO-VRPTW*) in terms of mean objective values ($\overline{f_1}$, $\overline{f_2}$) and mean runtime $\overline{\text{RT}}$

Type	*ACO-VRPTW* $\overline{f_1}$	*ACO-VRPTW* $\overline{f_2}$	*ACO-VRPTW* $\overline{\text{RT}}$/s	*IACO-VRPTW* $\overline{f_1}$	*IACO-VRPTW* $\overline{f_2}$	*IACO-VRPTW* $\overline{\text{RT}}$/s	relative error ϵ ϵf_1	ϵf_2	$\epsilon\overline{\text{RT}}$
25_1	4.29	396.47	1.08	4.27	396.76	0.93	**-0.57%**	0.07%	**-15.39%**
25_2	2.12	372.42	1.10	2.11	371.79	0.87	**-0.53%**	**-0.17%**	**-27.34%**
50_1	7.79	782.76	2.91	7.82	785.33	2.73	0.40%	0.33%	**-6.74%**
50_2	3.19	715.50	2.84	3.17	720.32	2.58	**-0.47%**	0.70%	**-10.31%**
100_1	15.15	1500.26	9.36	15.13	1495.02	8.68	**-0.14%**	**-0.35%**	**-7.91%**
100_2	5.59	1260.77	9.70	5.57	1257.13	9.15	**-0.53%**	**-0.29%**	**-6.00%**

6 Conclusions

In this paper, we proposed an initialized ACO – *IACO-VRPTW* – to solve vehicle routing problems with time windows. Our primary goal was to reduce the number of vehicles. We therefore adopt an elitist ANT-cycle model that awards ants finding schedules with fewer of vehicles, which may also accelerate the convergence speed.

In our experiments, we find that *IACO-VRPTW* can reach or reduce the vehicle numbers in 8 out of 18 problem types, at the cost of increasing the travel

distance slightly. For practical considerations, however, that small cost increase is far out-weighted by reduced personnel expenses. We find that our simple directional selection-based initialization procedure is effective in decreasing the vehicle number, total cost, and runtime.

In the near future, we will incorporate local search methods into our algorithm in order to refine the solutions found by the ants.

References

[1] Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: European Conference on Artificial Life, Paris, France, December 11-13, pp. 134–142 (1991)
[2] Yu, B., Yang, Z.Z., Yao, B.Z.: A hybrid algorithm for vehicle routing problem with time windows. Expert Systems with Applications 38(1), 435–441 (2011)
[3] Alvarenga, G.B., Mateus, G.R., de Tomi, G.: A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. Computer and Operations Research 34(6), 1561–1584 (2007)
[4] Homberger, J., Gehring, H.: Two evolutionary metaheuristics for the vehicle routing problem with time windows. INFOR 37, 297–318 (1999)
[5] Li, H., Lim, A.: Local search with annealing-like restarts to solve the vrptw. European Journal of Operational Research 150(1), 115–127 (2003)
[6] Berger, J., Barkaoui, M.: A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers and Operations Research 31(12), 2037–2053 (2004)
[7] Qi, C., Sun, Y.: An improved ant colony algorithm for vrptw. In: 2008 International Conference on Computer Science and Software Engineering, pp. 455–458 (2008)
[8] Homberger, J., Gehring, H.: A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. European Journal of Operational Research 162(1), 220–238 (2005)
[9] Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35(2), 254–265 (1987)
[10] Gambardella, L.M., Taillard, É., Agazzi, G.: Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. Technical Report IDSIA-06-99, IDSIA, Lugano, Switzerland (1999)
[11] Bent, R., Van Hentenryck, P.: A two-stage hybrid local search for the vehicle routing problem with time windows. Transportation Science 38(4), 515–530 (2004)
[12] Solomon, M.M.: Vrptw benchmark problems, `http://web.cba.neu.edu/~msolomon/problems.htm`

Deadline-Aware Event Scheduling for Complex Event Processing Systems*

Na Li and Qiang Guan**

Institute of Automation, Chinese Academy of Sciences,
95 Zhongguancun East Road, 100190 Beijing, P.R. China
qiang.guan@ia.ac.cn

Abstract. Complex event processing (CEP) plays an important role in developing responsive stream processing applications, with emphasis on "Velocity" from Big Data perspective. However, in emerging applications with heavyweight query requirements, the big rule set and the event coupling relationship could result in a complicated event graph. Meanwhile, state-of-the-art graph-based event processing approach employs the depth-first strategy and assumes that all complex events are equally important, which are more likely to reduce the scalability. In this paper, we focus on assigning priorities to complex events heuristically for the event graph model. The proposed priority dispatching policies take the following aspects into account: the topological particularity, the relative deadline and the dynamic process of event correlation, with the goal of minimizing the average response time. Furthermore, a priority-driven event scheduling strategy is presented, which aims to support real-time reasoning requirement. Finally, the experimental comparison with the standard graph-based event processing technique shows that the proposed event scheduling can yield a substantial improvement in the specific performances.

Keywords: Complex Event Processing, Event Graph, Heuristic Priority Scheduling, Deadline.

1 Introduction

Complex event processing denotes the algorithmic method for making sense of events by deriving higher-level knowledge (complex events) from large volumes of lower-level events in a timely fashion [1], and plays an important role in developing the responsive stream management system for Big Data applications. For instance, the rule-based Event-Condition-Action (ECA) paradigm is becoming useful in various areas, such as financial services, sensor networks, click stream analysis and so forth. The computation model of the ECA paradigm is the expressive event graph. However, when state-of-the-art graph-based event processing approaches [2–4] are applied in the emerging applications with heavyweight

* This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61174190, 61070048.

** Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 101–109, 2013.

query requirements, it brings about some new issues. The large rule set and the complex relationship between events could lead to a rather complicated event graph, and some critical event processing deadline constraints should be met.

Think for instance of the RFID-enabled hybrid manufacturing process. A number of sensors, including RFID tags and other transducers etc., are deployed to monitor the activity of the physical world. The distinguished feature of hybrid manufacturing process is that combining two or more technologies in one process or machine, which should result in a larger range of options for the creation and optimization of technology chains [5]. From the CEP perspective, it means that an event might trigger many parent complex events. If event graph is utilized, a rather complicated event graph will be attained, where residents multiple common nodes shared by different complex event branches. Also, the high level decision and control center can not suffer responding latency of queries too much. Therefore, the real-time performance issue from common nodes should be considered in this situation.

As for improving the real-time performance of the CEP systems, the coverage of this issue in the literature is quite limited. Paper [6] presents an event processing engine to deal with the large volumes of RFID data by using buffer to pre-load data for improving the efficiency and decreasing the response time. Both [7] and [8] improve the real-time capability of CEP systems from the perspective of implementing. Active CEP technology in [8] focuses on embedding active rule support. The key concept in [9] is subdividing the processing task into basic stream operators and then scheduling the operator processing tasks so as to minimum processing delay. Nevertheless, this method is devoted to DSMS. Few of implementations of graph-based CEP framework have taken optimizing the time consumption into consideration, especially for the event graph with many common nodes. The algorithm in [10] is based on heuristic search for an event graph However, the implementation detail is omitted and the focus is the action component of the ECA paradigm.

In this paper, we mainly focus on improving real-time event detection capability for the complicated event graph generated from multiple coupled rules. By investigating the topological particularity of the event graph and extending to the deadline-aware model, we propose a heuristic priority event scheduling strategy. The priority assignment to each complex event relies on the following factors: the topological particularity, the relative deadline and the dynamic process of event correlation. On the basis of the proposed priority scheduling strategy, we expand the classical graph-based event detection method by integrating the deadline-aware mechanism. Experimental evaluation results are presented to demonstrate the effectiveness of the proposed method.

2 Deadline-Aware Event Model

2.1 Events

Generally speaking, an event is an atomic (happens completely or not at all) occurrence [1]. Event types specify the event structures, whereas event instances

indicate the occurring of events defined by the specific event type. We will use E (upper case alphabets) to represent an event type and e (lower case alphabets) for an instance of E. Event can be broadly classified into the primitive event and the complex event.

Primitive Event is pre-defined in the system and can be sensed directly. One notion of the primitive event is $E_a =< SID, eventType, [t_{start}, t_{end}] >$ [4]. SID identifies the event source. $[t_{start}$,$t_{end}]$ specifics the occurrence time of E_a's instance. Generally, for the primitive event, $t_{start} = t_{end}$.

Complex Event is formed by applying a set of operators to primitive or complex events. There are two types of event operators: logical and temporal. In this work, we adopt event operators in [3]. The detailed definitions and semantics for event operators can be found in [3].

2.2 Rules

Complex events are processed according to rules. The rule formulation in this work is akin to that defined in standard ECA paradigm [2, 3], where clause *Event Pattern* formalize the desired complex event. In addition, optional clause *DEADLINE* is introduced to represent the timing constraint on this rule.

2.3 Event Graph

Complex events are detected based on the event graph, which is a tool representing both event operation and event relationship. Event graph consists of multiple event trees with common event nodes, where leaf nodes refer to primitive events and root nodes correspond to complex events defined by *Event Pattern* in rules. For convenience, constituent events except primitive events are called mediate events (mediate nodes in event graph). Event instances and parameters are broadcasted from leaf nodes upwards. The diagram of the event tree is shown in Figure 1(a). In the diagram, the mediate event E_c is expressed by a circle and the primitive event E_a is referred as a rectangular. P_k, as the root complex event of the event tree, would trigger the action further and is labeled by a filled circle. In this paper, we mainly focus on the problem of adjusting the processing order of event processing tasks triggered by the common event node simultaneously.

2.4 Deadline

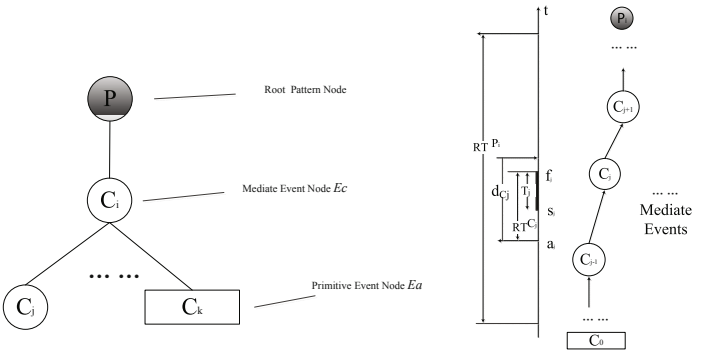
Based on Real-time System theory [11], the deadline in our model is classified into two kinds.

i) The definitions related to the deadline of the root pattern P_i:

Response Time of Root Pattern RT^{P_i} denotes the time period from the occurrence to the responding (detection) of P_i's instance.

Deadline of Root Pattern d_{P_i} is straightforward and can be filled with *DEADLINE* part defined in the rule which P_i belongs to.

ii) The definitions related to the deadline of the mediate event C_j : (as shown in Figure 1(b)



(a) Event tree diagram (b) Deadline parameters

Fig. 1. Event tree and deadline

Arrival Time a_j is the release time of the event processing task of the mediate event C_j. The processing task of C_j is released when the child processing task of C_{j-1} has just finished processing.

Response Time RT^{C_j} is the time interval between arrival time a_j and finishing time f_j.

Affection Set Γ_j is the set of root patterns, whose constituent includes C_j.

Affection F_j equals the cardinality of the affection set Γ_j; $F_j = |\Gamma_j|$.

Deadline of Mediate Event is different from that of root pattern nodes because it is dynamic and depends on deadlines of root patterns as well as the complex event processing procedure. As for a mediate event C_j, upper bound of deadline can be calculated. For a root pattern P_i in the C_j's affection set Γ_j, $d_{P_i,j}$ denotes the deadline upper bound of C_j versus P_i.

$$d_{P_i,j} = d_{P_i} - (a_j - t_{end}(P_i))$$

Hence, d_{C_j}, the deadline upper bound of C_j, can be attained using the following formula.

$$d_{C_j} = \min_{\Gamma_\mathrm{j}} \{d_{P_i,j} | P_i \in \Gamma_\mathrm{j}\}$$

2.5 Problem Formulation

In graph-based CEP framework, complex event detection is essentially a recursive graph traversal process. However, when confronted with a complicated event graph with multiple common nodes, current methods ignore the time pressure of different patterns and overlook the event scheduling issue from common nodes. In order to characterize the differences in response time urgency among complex events, we extend the current framework by introducing the deadline concept. In the deadline-aware CEP system, the goal of event scheduling is not only to improve throughout, i.e. to produce valid complex events as many as possible, but

also to make the complex events meet their deadline as far as possible. Therefore, the event scheduling can be formalized as the following optimization problem.

Given a set of business rules, the corresponding **Root Pattern Set** Γ and event graph G are immediate. For a primitive event stream $[a_1, a_2, \cdots, a_n]$, find out an optimal detected event instance stream (non-leaf event instance node sequence in G generated from event detection processing through traversing graph G in a bottom-up manner) $[v_1, v_2, \cdots, v_m]$, whose length m is not fixed, satisfies:

$$\begin{aligned} &\max \textstyle\sum_{i=1}^{m} \mu_\Gamma(v_i) \\ s.t. \quad &\mu_\Gamma(v_i) * RT^{v_i} \leq d_{v_i} \quad i = 1 \ldots m \end{aligned} \tag{1}$$

Here, $\mu_\Gamma(x)$ is the characteristic function on set Γ, i.e. $\mu_\Gamma(x) = 1, x \in \Gamma$, otherwise $\mu_\Gamma(x) = 0$.

Obviously, it is difficult to directly solve Formula (1) online due to the features of event stream. From computation efficiency perspective, Formula (1) can be relaxed to the problem that aims to minimizing the average response time of detected complex events.

3 Heuristic Priority Event Scheduling

Based on the previous analysis, we propose a heuristic priority event scheduling strategy to deal with multiple complex event processing tasks triggered simultaneously. The priority evaluation underlies the event scheduling strategy.

3.1 Priority Assignment Policies

Similar to other task scheduling algorithms [11], the heart of our proposed method remains how to assign the priority to each scheduled event. In this section, we present the following policies for assigning priorities to complex events that account for both event graph topological characters and response deadlines of complex events.

Policy 1 *Node Degree First*

Intuitively, the mediate event node with more adjacent nodes is more complex and maybe trigger more events. Therefore, it should have higher priority. To reflect this simple idea, a relevant priority metric Pr^{degree} can be introduced, and $Pr^{\text{degree}}(C_j) = DN_{C_j}$. Here, DN_{C_j} denotes the degree of the event node C_j in the event graph G.

Policy 2 *Node Affection First*

The main idea behind this policy is that of giving higher priorities to event nodes, which have larger affection set. This should increase the likelihood of improving the throughout. The corresponding priority metric can be defined as $Pr^{\text{affection}}(C_j) = |\Gamma_j|$.

Policy 3 *Node Deadline First*

As the routine task scheduling approaches, the concept of deadline is adopted in that the deadline reflects the urgency of completing event processing task.

Slightly different with the definition in the regular scheduling methods, the deadline for mediate events can not be gained in a priori. See the discussion in last section. $Pr^{\text{deadline}}(C_j) = d_{C_j}$.

The presented policies are simple, heuristic and can be used in complex event processing methods easily. Moreover, some direct arithmetic computations show these policies contribute to reducing response time under some assumptions. Due to space limitations, the detailed analysis is omitted.

3.2 Priority-Based Event Processing Strategy

By means of the above priority metrics, a general priority evaluation model is generalized to act as an important basis for deadline-aware event processing systems. To cover all advantages of three metrics, the weighted combination arithmetic is used. The priority evaluation $Pr(C_j)$ for the event node C_j is as follows.

$$\begin{aligned} Pr(C_j) = (\alpha Pr^{\text{affection}}(C_j) + \beta Pr^{\text{degree}}(C_j))^{\frac{\alpha+\beta}{2}} \\ + Pr^{\text{deadline}}(C_j)(1 - \tfrac{\alpha+\beta}{2}) \end{aligned} \tag{2}$$

Here, $\alpha, \beta \in [0, 1]$ and $\alpha + \beta = 1$. If $\alpha = 0, \beta = 1$, then $Pr = Pr^{degree}$. If $\alpha = 1, \beta = 0$, then $Pr = Pr^{affection}$. If $\alpha = 0, \beta = 0$, then $Pr = Pr^{deadline}$.

The architecture of integrating the priority based event scheduling with CEP framework is shown in Figure. 2. Collectively, Priority Assignment module and Event Scheduling Strategy Set constitute Priority-based Complex Event Detection Strategy. The evaluation polices used in the priority assignment process are defined according to the event query set and Complex Event Detection model. As to the simultaneous queries upon common event nodes, we enhance the graph-based CEP framework with priority-based scheduling strategy. The implemental details associated to other modules in Figure. 2 can be found in [3].

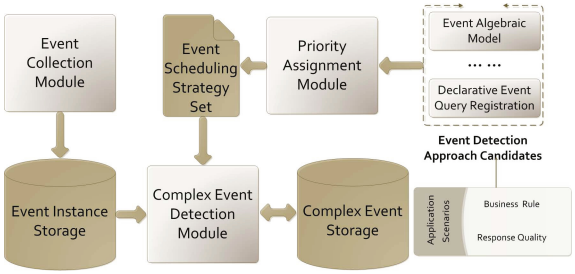


Fig. 2. The extended CEP framework with embedded event scheduling strategy

4 Evaluation

In this section, we present the result of the proposed method in the context of the complicated event graph with multiple common nodes. We augmented the

current graph-based CEP framework (GEP) to add support for priority-based event scheduling, called PS-GEP method.

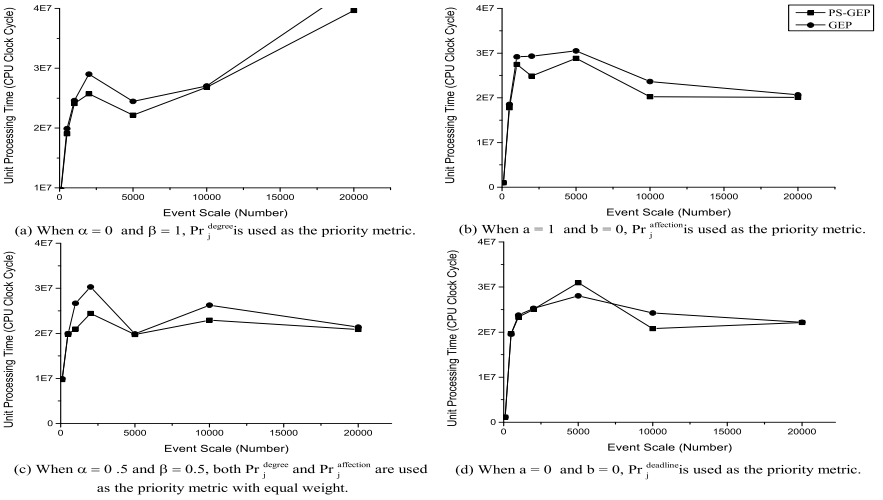
Firstly, it is not easy to valuating the performance of the GCEP framework because it is strongly influenced by the workload and there are no publicly available data sets for evaluating. Thus, we use a large number of synthetic workloads. Primitive event instances are generated randomly. For complex root patterns, a set of the simultaneous queries is introduced to illustrate the comparison between GEP and PS-GEP. Such multiple simultaneous queries capture the common relationships among events by operators SEQ+(E_i), OR(E_i;E_j) and TSEQ+(E_i,t_1,t_2) [3] and listed in Table 1. Secondly, for the primitive event processing cost, we used the *Unit Processing Time* (UPT) [12] as an evaluation metric; it is defined as t_{total}/N_{toal}, where N_{total} is the total number of the input primitive event instances, and t_{total} is the total elapsed processing time, not including time to deliver the output and make decisions. For the root pattern processing cost, we measured average *Response Time* (RT) (defined in subsection 2.5) of root complex patterns; it is defined as $\sum RT^{P_i}/N_{ComplexInstances}$, where $\sum RT^{P_i}$ is the sum of the RT of all monitored root patterns' instances, and $N_{ComplexInstances}$ is the total number of root pattern instances monitored.

Table 1. A Simultaneous Query Scenario

	Parameters in the base scenario
1	TSEQ+(WITHIN(SEQ+(WITHIN(SEQ+(OR(E_2;E_0)),t_1)),t_2),t_3,t_4)
2	TSEQ+(WITHIN(SEQ+(OR(E_1;WITHIN(SEQ+(OR(E_2;E_0)),t_1))),t_2),t_3,t_4)
3	OR(E_1;WITHIN(SEQ+(OR(E_2;E_0)),t_1))
4	OR(E_0;WITHIN(SEQ+(E_2),t_1))
5	TSEQ+(OR(E_1;WITHIN(SEQ+(OR(E_2;E_0)),t_1)),t_2,t_3)

4.1 Primitive Event Processing Performance

In this experiment, PS-GEP with four suits of {α, β} are investigated, including $\alpha = 0$, $\beta = 1$; $\alpha = 1$, $\beta = 0$; $\alpha = 0.5$, $\beta = 0.5$; $\alpha = 0$, $\beta = 0$. Each configuration represents a typical priority assignment case. PS-GEP takes advantage of the scheduling strategy for adjusting the execution order of event processing, when multiple processing tasks are triggered simultaneously. PS-GEP is compared against GEP under different scales of event sources in Figure 3. It shows the UPT of PS-GEPs and GEP in simultaneous event sources scenario. Each pair PS-GEP and GEP was executed for timely independent. Thus, the performance of GEP looks different among all comparisons. However, we observe that PS-GEP of four sets of {α, β} perform slightly better than GEP over UPT. Figure 3(c) shows that $\alpha = 0.5$, $\beta = 0.5$ is kind of balance metrics. From the perspective of priority metric definition, Figure 3 indicates that $Pr^{affection}$ tends to order processing tasks globally.



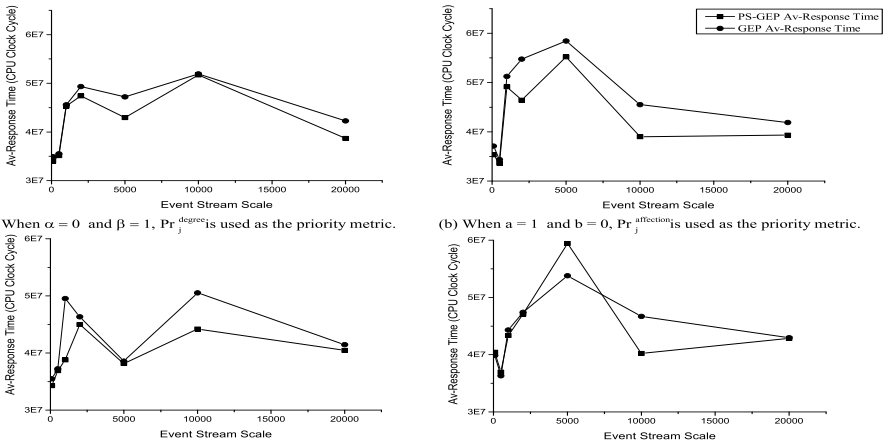
(a) When $\alpha = 0$ and $\beta = 1$, Pr_j^{degree} is used as the priority metric.

(b) When a = 1 and b = 0, $Pr_j^{affection}$ is used as the priority metric.

(c) When $\alpha = 0$ .5 and $\beta = 0.5$, both Pr_j^{degree} and $Pr_j^{affection}$ are used as the priority metric with equal weight.

(d) When a = 0 and b = 0, $Pr_j^{deadline}$ is used as the priority metric.

Fig. 3. Primitive Events Performance



(a) When $\alpha = 0$ and $\beta = 1$, Pr_j^{degree} is used as the priority metric.

(b) When a = 1 and b = 0, $Pr_j^{affection}$ is used as the priority metric.

(c) When $\alpha = 0$ .5 and $\beta = 0.5$, both Pr_j^{degree} and $Pr_j^{affection}$ are used as the priority metric with equal weight.

(d) When a = 0 and b = 0, $Pr_j^{deadline}$ is used as the priority metric.

Fig. 4. Complex Events Cost

4.2 Root Pattern Processing Cost

As to the evaluation of root pattern detection cost, we inspect the average Response Time of the multiple simultaneous query set defined in Table 1. The four comparisons are running on the random input streams respectively. In each case, there exists a dominate factor in priority dispatch polices. Thus, in practice, priority-based scheduling tends to speedup part of event processing instead of all. However, Figure 4 demonstrates that PS-GEP presents some improvement in RT averagely. Figure 4(a)(b)(c) perform better by considering the topology characteristics to balance the idle event processing time.

5 Conclusions

In this paper, we concentrate on addressing the issue of efficiently monitoring complex event streams by graph-based CEP technique. In particular, the complicated event graph with multiple common nodes shared by complex events is the research focus. To model the temporal characteristic of the complex event, we extend the graph-based complex event framework by introducing the deadline concept. Furthermore, three priority assignment policies are presented to cover the event graph topological feature and the relative deadline of the complex event. On that basis, heuristic priority-driven event scheduling strategy is proposed and integrated into the current graph-based CEP method. Finally, computer simulation results confirm the superior performance of proposed event scheduling strategy for typical concurrent multiple queries.

References

1. Luckham, D.C.: The Power of Events. Addison-Wesley Longman Publishing Co., Boston (2001)
2. Adaikkalavan, R., Chakravarthy, S.: SnoopIB: Interval-Based Event Specification and Detection for Active Databases. Data and Knowledge Engineering 59, 139–165 (2006)
3. Wang, F., Liu, S., Liu, P.: Complex RFID event processing. VLDB Journal 18, 913–931 (2009)
4. Seven, H., Alexandra, P., Fatose, X.: Reasoning in Event-Based Distributed Systems. Springer, Massachusetts (2011)
5. Nau, B., Roderburg, A., Klocke, F.: Ramp-up of hybrid manufacturing technologies. CIRP Journal of Manufacturing Science and Technology 4, 313–316 (2011)
6. Yan, L., Dong, W.: Complex Event Processing Engine for Large Volume of RFID Data. In: Proceedings of 2nd International Workshop on Education Technology and Computer Science, pp. 429–432 (2010)
7. Magid, Y., Adi, A., Barnea, M., et al.: Application Generation Framework for Real-Time Complex Event Processing. In: Proceedings of 32nd Annual IEEE ICSA, pp. 1162–1167 (2008)
8. Wang, D., Rundensteiner, E., et al.: Active complex event processing: applications in real-time health care. In: Proceedings of 38th VLDB, pp. 1545–1548 (2010)
9. Schmidt, S., Legler, T., Schaller, D., et al.: Real-Time Scheduling for Data Stream Management Systems. In: Proceedings of the 17th ECRTS, pp. 167–176 (2005)
10. Qiao, Y., Li, X., Wang, H., Zhong, K.: Real-Time Reasoning Based on Event-Condition-Action Rules. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 1–2. Springer, Heidelberg (2008)
11. Buttazzo, G.: Hard Real-time Computing Systems. Springer, Berlin (2005)
12. Lee, S., Lee, Y., Kim, B., et al.: High-Performance Composite Event Monitoring System Supporting Large Numbers of Queries and Sources. In: Proceedings of the Fifth ACM International Conference on Distributed Event-Based Systems (DEBS 2011), pp. 137–148 (2011)

A Discrete Hybrid Bees Algorithm for Service Aggregation Optimal Selection in Cloud Manufacturing

Sisi Tian[1,2], Quan Liu[1,2], Wenjun Xu[1,2], and Junwei Yan[1,2]

[1] Key Lab. of Fiber Optic Sensing Technology and Information Processing of Ministry of Education, Wuhan 430070, China
[2] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China
{tiansisi,quanliu,xuwenjun,junweiyan}@whut.edu.cn

Abstract. Facing to the globalization and increasing competition of manufacturing enterprise, how to integrate the existent manufacturing services in cloud manufacturing model to form the newly value-added services in order to fulfill the user requirements has become a significant issue in manufacturing area. In this context, a discrete hybrid Bees Algorithm (DHBA) is proposed to solve service optimal the selection in resource service aggregation. The problem of service aggregation with QoS global optimal is transformed into a multi-objective services aggregation optimization with QoS constraints, and DHBA is utilized to produce a near-optimal solution. A case study together with a set of simulation experiment is presented and the results demonstrate the effectiveness and feasibility of the proposed method.

Keywords: Cloud manufacturing, service aggregation, Bees Algorithm, Pareto optimal.

1 Introduction

The aggregation and synergy ability of manufacturing resources have become the key elements of enterprise competitiveness. Cloud Manufacturing (CMfg) [1] is a requirements-driven, service-oriented manufacturing new model which aims at realizing the sharing and on-demand using of manufacture resources and manufacture capabilities, so as to improve the utilization efficiency of manufacturing resource service. Cloud service aggregation is one of the key approaches to achieve resource value-added which playing an important role in CMfg implementation process. When a user's request cannot be satisfied by any individual service, there should be a possibility to combine the existing cloud services together so as to fulfill the customer's requirements. Since a number of services perform similar functions, there must be many aggregation services which can achieve user's requirements. Therefore, how to select the best one from all possible combinations to complete the task is the chief task, i.e., service aggregation optimal selection.

Service aggregation selection is a multi-objective optimization problem with QoS constraints and belongs to an NP-Complete problem. The existing intelligent

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 110–117, 2013.

optimization algorithms such as Genetic Algorithm [2], Swarm-based optimization algorithms (SOAs) including Particle Swarm Optimization [3] that find multiple Pareto-optimal [4] solutions have been proposed to solve multi-objective problem. Bees Algorithm (BA) [5] is a swarm-based optimization algorithm which performs a combination of neighborhood search with random global search inspired by the way forage food of honey bees. According to the extensive evaluate results in [6], we know that BA outperforms other SOAs in terms of learning accuracy and convergence speed. BA has applied to different problems including job scheduling, manufacturing cell and multi-objective optimization problems [7]. Due to the aforementioned advantages, BA has a significant potential for solving the optimization problems. DHBA using a combination of Pareto optimal with BA is presented in the paper. The research on BA for solving service selection problem is still relative limited. A well-designed crossover [8] operator and reverse [9] strategy is adopted to enhance the exploitation and exploration capability of algorithm, annealing mechanism of acceptance criteria to prevent algorithm from local solution.

2 Service Aggregation Model

Suppose there are n services $(S_1, S_2, \cdots, S_n)$ in a composite service, S_i is the i^{th} component service of an aggregation service. Each component service has a set of candidate services with similar function, QoS attributes should be considered to differentiate service. We consider four quality properties: execution cost (C), response time (T), availability (Avl) and reliability (Rel). Four basic control structures are showed in Fig.1. Aggregation functions are defined in Table 1 [10] to calculate the quality of an aggregation service. p_i is the probability of being chosen of i^{th} service in conditional structure, and $\sum_{i=1}^{n} p_i = 1$, k is the executions of a service in loop structure.

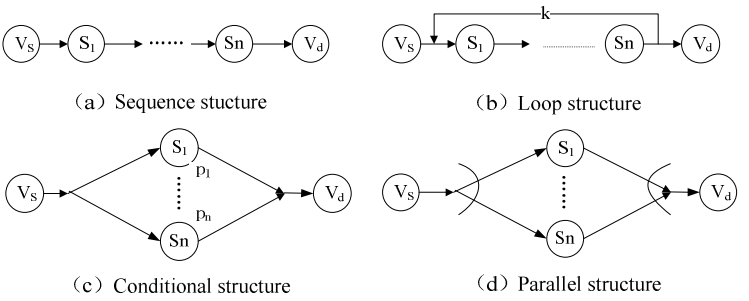


Fig. 1. Four basic control structures

Table 1. QoS aggregation functions

QoS attribute	Sequence	Parallel	Conditional	Loop
Execution cost(C)	$\sum_{i=1}^{n} C(s_i)$	$\sum_{i=1}^{n} C(s_i)$	$\sum_{i=1}^{n} p_i C(s_i)$	$k\sum_{i=1}^{n} C(s_i)$
Response time(T)	$\sum_{i=1}^{n} T(s_i)$	$Max\left\{T(s_i)_{i\in\{1,\ldots,n\}}\right\}$	$\sum_{i=1}^{n} p_i T(s_i)$	$k\sum_{i=1}^{n} T(s_i)$
Availability(Avl)	$\prod_{i=1}^{n} A(s_i)$	$\prod_{i=1}^{n} A(s_i)$	$\prod_{i=1}^{n} p_i A(s_i)$	$\prod_{i=1}^{k}\prod_{i=1}^{n} A(s_i)$
Reliability(Rel)	$\prod_{i=1}^{n} R(s_i)$	$\prod_{i=1}^{n} R(s_i)$	$\prod_{i=1}^{n} p_i R(s_i)$	$\prod_{i=1}^{k}\prod_{i=1}^{n} R(s_i)$

3 Discrete Hybrid Bees Algorithm

3.1 Algorithm Description

During the establishment of service aggregation model, DHBA is designed to look for feasible solution. The main steps are depicted in Fig. 2.

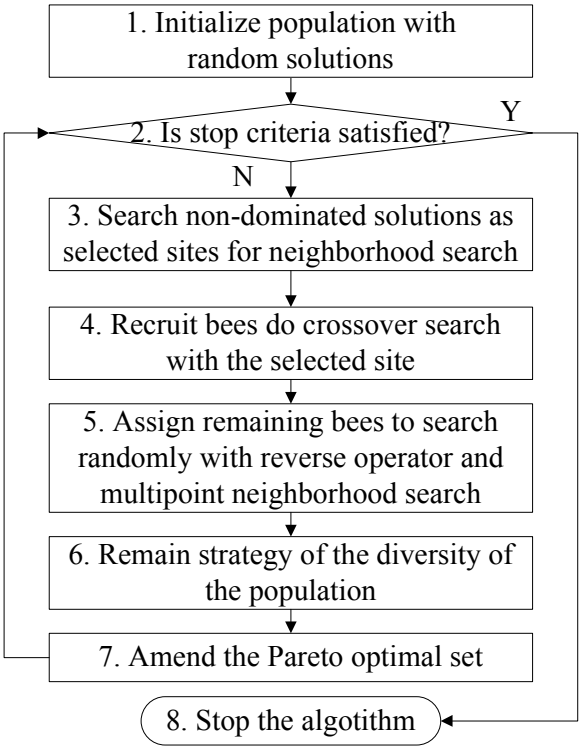


Fig. 2. The main steps of DHBA in service aggregation selection

In step 1, n scout bees which satisfy QoS constraints are randomly distributed in the solution space. In step 3, non-dominated bees are chosen as "selected sites" and for neighborhood search. In step 4, some honey bees will be selected as recruit bees which do neighborhood search around selected site. Neighborhood search around the selected site can guide the search towards the Pareto-optimal set. The remaining bees use multipoint search and reverse operator to produce new potential solutions in step 5. In step 6, a simple and effective way is used to keep the population diversity in order to achieve a well distributed spread trade-off front. The individual position is different from each other in each generation. If there are some bees in the same position, we reserve an individual's information and then initialize the remaining individuals. Step 8 makes it possible for BA to deal with multi-objective optimization problem. Steps 3 to 7 are repeated until the termination criteria are satisfied.

3.2 Coding Strategy

Each service node in process of service aggregation is ordered in sequential. Every aggregation service is encoded as a bee, and integer coding is adopted to encode the candidate services so as to map the aggregation service into position vectors. As showed in Fig. 3. $S(i)$ represents the code of component service selected from the corresponding candidate services of service node S_i, $i \in \{1,\ldots,10\}$.

S(1)	S(2)	S(3)	S(4)	S(5)	S(6)	S(7)	S(8)	S(9)	S(10)

Fig. 3. Coding strategy of bee individual

3.3 Crossover Operator: Recruit Bee

Half of the population except selected sites is selected as recruit bees. Each recruit bee random selects a selected site and performs crossover operator with the selected site. Given a selected site $C1$ and a recruit bee $C2$ in Fig.4.The steps of the crossover operator are as follows:

Step 1: Random generate two numbers i and j ($i < j, i, j \in \{1,\ldots,m\}$), and select the subsection between i and j.

Step 2: Produce a new solution named $C3$ by coping the selected subsection from $C1$ to the corresponding position of $C2$.

By introducing the crossover operator, recruit bees inherit part of good information from the selected site.

3.4 Reverse Operator

The remaining individual except recruit bees and selected sites is selected to search randomly with a combination of multipoint search with reverse strategy at an equal

probability. Multipoint search is an extension of the single point search. Given a scout bee $C1$ in Fig.5, the steps of the reverse operator are as follows:

Step 1: Random generate two numbers i and j ($i < j$, $i, j \in \{1, \ldots, m\}$), and select the subsection between i and j.

Step 2: Produce a new solution named $C2$ by reversing coding the subsection.

The search strategy of remaining bees is that generate a random number p ($p \in (0,1)$), if $p < 0.5$, multipoint search strategy is adopted to produce $C2$, otherwise, reverse operator to produce $C2$.

The reverse and multipoint strategies avoid local optima and increase the population diversity.

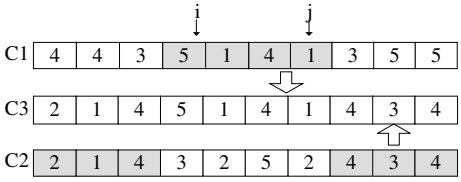


Fig. 4. Crossover operator

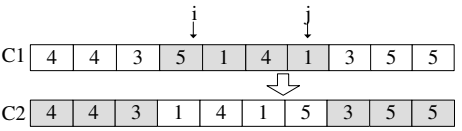


Fig. 5. Reverse operator

3.5 Annealing Strategy

In order to preserve the population diversity and prevent to stick on a local solution, we introduce an annealing strategy to accept the inferior solution at a certain probability. Acceptance probability of the solution in the i^{th} iteration is defined as follows.

$$p = P_{\max} \cdot e^{-0.02i} \tag{1}$$

$P_{\max}$ is the max value of p, and $P_{\max}$ =0.3. p changes as the number of iterations increasing. In the early evolution, a greater probability is given to accept inferior solutions so that the algorithm can prevent premature convergence; in the late stage of the evolution, p gradually becomes 0 to ensure the algorithm convergence. According equation (1), we generate a probability p in each iteration. For new solution produced in the process of neighborhood search and random search, p is used to accept inferior solution.

4 Case Study and Performance Evaluation

In this section, an aggregation service, motorcycle design and assembly application is given to verify the feasibility and effectiveness of DHBA. The main manufacturing process shown in Fig.6, and where S_1 represents 3D model design, S_2 product style design, S_3 frame assembly, S_4 front portion assembly, S_5 engine assembly, S_6

electrical installation, S_7 braking installation, S_8 electrical harnesses installation, S_9 product assembly, S_{10} product test. Node S_3 - S_8 are parallel structure. Each node can be provided by a number of candidate services. Execution cost and response time are optimization objectives. Global constraints are that maximum execution cost ($C_{\max}$) and maximum response time ($T_{\max}$) of an aggregation path, local constraints are minimum availability ($A_{\min}$) and reliability ($R_{\min}$). The mathematical model is show as follows:

$$\text{Minimum} \quad \begin{cases} T(S) = T(S_1)+T(S_2)+T(S_9)+T(S_{10})+Max(T(S_i)_{i\in\{3,\ldots,8\}}) \\ C(S) = \sum_{i=1}^{10} C(S_i) \end{cases} \tag{2}$$

$$\text{s.t.} \quad \begin{cases} \mathrm{Re}l(S_i) - \mathrm{Re}l_{\min} \geq 0, i \in \{1,\ldots,10\} \\ Avl(S_i) - Avl_{\min} \geq 0, i \in \{1,\ldots,10\} \\ C_{\max} - C(S) \geq 0 \\ T_{\max} - T(S) \geq 0 \end{cases} \tag{3}$$

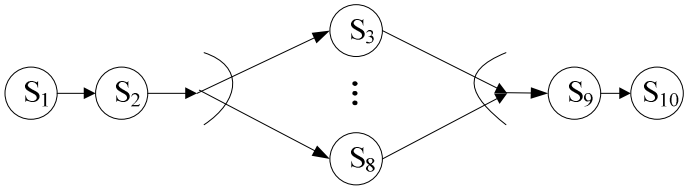


Fig. 6. An example of service aggregation S

The value of each QoS attribute for each candidate is randomly generated in a range as shown in Table 2, For simplicity, we assume that each abstract service contains the same number of candidate services, and $C_{\max}$ =500, $T_{\max}$ =30, $A_{\min}$ =0.6, $R_{\min}$ =0.6. We utilize Matlab to implement the proposed algorithms.

Table 2. The range of the value of each QoS value

QoS parameter	Range
Execution cost	[1,100]
Response time	[1,10]
Availability	[0.5,1]
Reliability	[0.5,1]

The number of initial population is 100. When the number of candidate services is 5, 10, 20, each situation is executed 10 times in generations 100, 200, 300, 400, the result is showed in Fig.7. When the number of candidates is 20 and generations is 400, the execution time is 8s, the result shows the effectiveness of the proposed algorithm, which can performance well to solve many actual aggregation problems.

The Pareto optimal set is showed in Fig.8 in which the number of candidates is 20 and generations is 400, which shows the relationship between execution cost and response time of aggregation service, from the experimental results, we can see that the non-dominated solutions achieve a well distributed, wide spread trade-off front to ensure the diversity of the population.

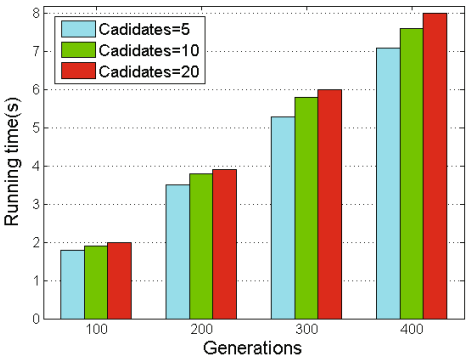


Fig. 7. Average running time of different generations

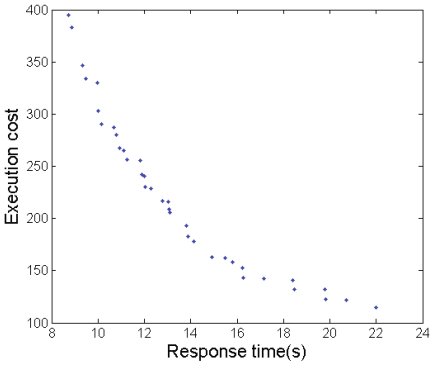


Fig. 8. Pareto optimal set distributed

When the number of candidates is 5 and the generations are 400, Fig.9 shows the algorithm by adopting the annealing mechanism performs better than the algorithm which doesn't adopt annealing mechanism. DHBA can generate better quality of solution, and execution cost is less in same response time. Furthermore, as the number of initial population n is 100, the number of candidates is 10 and the generations is 400, we compared the proposed algorithm with multi-objective Genetic Algorithm (MOGA) in [2] which can solve web service selection algorithm, crossover probability of MOGA is 1, mutation probability is 0.15. Computational results show that DHBA can find more effective and better Pareto optimal solutions.

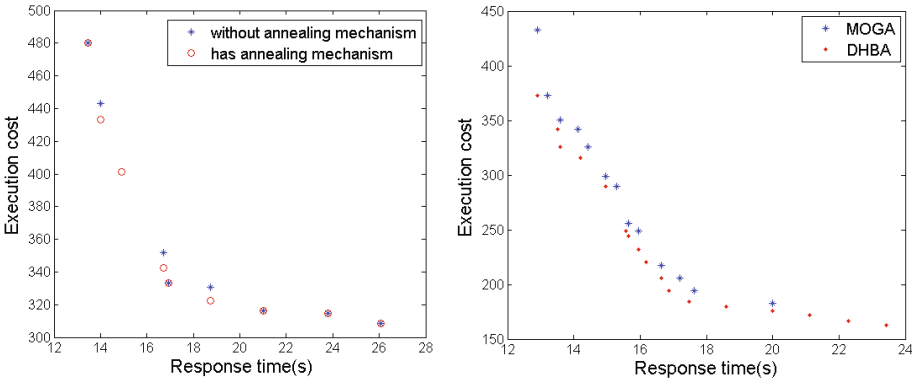


Fig. 9. Comparison of Pareto optimal set distributed

5 Conclusions

In the paper, we have proposed an efficient algorithm for solving service selection problem in CMfg. Experimental results on a case study show that the algorithm performs better and is more competitiveness than other existing related algorithms.

In future, we will study the service correlations in service aggregation and establish a correlation-aware QoS model, and thus the algorithm proposed in the paper would be improved to solve more complex and the large scale services aggregation problem in CMfg.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant No. 51175389, Key Project of Chinese Ministry of Education under Grant No. 313042, and the Fundamental Research Funds for the Central Universities under Grant Nos. 2013-VII-001 and 2013-IV-046.

References

1. Zhang, L., Luo, Y.L., Tao, F., Li, B.H., Ren, L., Zhang, X.S., Guo, H., Cheng, Y., Hu, A.R., Liu, Y.K.: Cloud manufacturing: A new manufacturing paradigm. In: Enterprise Information Systems, pp. 1–21 (2012)
2. Liu, S.L., Liu, Y.X., Zhang, F., Tang, G.F., Jing, N.: A dynamic web service selection algorithm with QoS global optimal in web services composition. Journal of Software 18(3), 646–656 (2007)
3. Tao, F., Zhao, D.M., Hu, Y.F., Zhou, Z.D.: Correlation-aware resource service composition and optimal-selection in manufacturing grid. European Journal of Operational Research 201(1), 129–143 (2010)
4. Arora, J.S.: Introduction to optimum design. Elsevier, New York (2004)
5. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University: Cardiff (2005)
6. Pham, D.T.: The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci. 233(12), 2919–2938 (2009)
7. Pham, D.T., Ghanbarzadeh, A.: Multi-objective optimisation using the bees algorithm. In: Innovative Production Machines and Systems Conference, IPROMS 2007 (2007)
8. Goksala, F.P., Karaoglanb, I., Altiparmak, F.: A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Computer & Industrial Engineering 65(1), 39–53 (2012)
9. Nagy, G., Salhi, S.: Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research 162(1), 126–141 (2005)
10. Zeng, L.Z., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004)

Continuous Motion Recognition Using Multiple Time Constant Recurrent Neural Network with a Deep Network Model

Zhibin Yu and Minho Lee*

School of Electronics Engineering Kyungpook National University,
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Republic of Korea
{ahriman1985abr,mholee}@gmail.com

Abstract. Multiple timescale recurrent neural network (MTRNN) model is a useful tool to model a continuous signal for a dynamic task such as human action recognition. Different setting of initial states in the MTRNN brings us convenience to predict multiple signals using the same network model. On the contrary, optimal switching for suitable initial states in the slow context unit of the MTRNN becomes critical condition to achieve desired multiple dynamic tasks. In this paper, we propose a hybrid neural network model combining the MTRNN with a deep learning neural network (DN), which is to overcome the problem related to the initial state setting in the MTRNN. The DN together with MTRNN generates a suitable initial state for the slow context units in the MTRNN according to automatically detected situation change. We apply our approach to 20 motion skeleton units, which is obtained by KINECT, to construct three kinds of human motion sequences. The results show that the proposed method is able to recognize various motions using proper initial state information in a real-time procedure.

Keywords: Motion recognition, MTRNN, Classification, Deep learning neural network, hybrid neural network.

1 Introduction

In recent, video based motion recognition has been a hot research topic owing to the development of sensor systems such as Microsoft KINECT [1]. It can provide a new way to design an advanced human computer interface system [2], and also open a new business model for video based surveillance system [3]. Although human observers can easily recognize the activities of other humans, it's still a challenging task via mathematical and computational algorithms even when we use the advanced sensor systems. Vision-based recognition is proved to be an efficient method on human motion recognition [4]. Based on vision information, human motion can be roughly divided into static posture recognition and dynamic process of recognition [5]. Researches based on the static posture recognition focus on the shapes or postures

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 118–125, 2013.

represented by a single image [6], while researches related with dynamic process try to use dynamic models such as Hidden Markov Model to recognize the motion sequences [7]. The effectiveness of Deep Learning Neural Networks (DN) proposed by Fukushima [8] and extended by Hinton [9] demonstrates good performance on static image interpretation [10] and can effectively handle static posture recognition and evaluation.

Yamashita and Tani [11] proposed a neuro-dynamic model referred to multiple timescale recurrent neural network (MTRNN) to imitate the human motor control system. As an extension of Recurrent Neural Networks (RNN) [12], MTRNN has an important feature called "self-organization", which is a phenomenon that a global coherent structure appears in a system not by a central authority but by local interaction among elements of the system [13]. Those features of MTRNN make it suitable for dynamic motion recognition.

Initial states of MTRNN bring convenience to us when we want to estimate various kinds of signal sequences using the same network model. But, we have to assign a proper initial state for a suitable start signal to get a desired sequence. If we want to predict other signals, we have to assign new initial state to each signal. If we use an incorrect initial state to estimate other signals, the prediction error will increase. In general, the conventional MTRNN model is unable to determine which initial states are proper in a given condition. Also, it is difficult to estimate optimal switching time of predefined initial states according to a situation change.

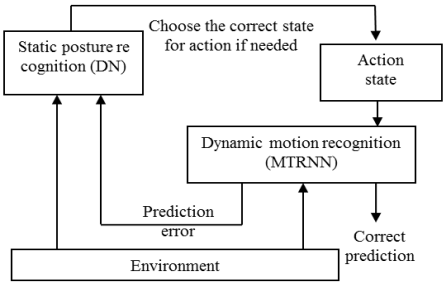


Fig. 1. Overview of the hybrid system

In this paper, we combine the static posture recognition system and the dynamic motion recognition system together and expect them to cooperate with each other to solve the problem of automatic setting of initial states according to a situational change. Fig. 1 shows the proposed architecture which hybrids the MTRNN with DN to automatic setting of initial states. We use a DN using error back propagation (EBP) learning algorithm to build the static posture recognition model and MTRNN using back propagation through time (BPTT) learning algorithm to construct the dynamic motion recognition model. We collected 40 dimension data of human motion based on skeleton information using KINECT. The prediction error of the 40 dimension data will be additionally used as a part of input data of the DN. The difference between measurement data and prediction values in each dimension is also used as another part of input data of the DN. The output of DN will instantly show the class label of initial

states based on the current signal. That means the proposed model can recognize the human action immediately and needn't wait until the action is finished.

2 Related Works

In MTRNN, Self-organizing map (SOM) [14] is used as a unsupervised learning method to accept vision signals and predicts the output signals. Since we use a virtual robot in our experiments, we simplified the model proposed by Yamashita et al. [11] and employ an 8*8 map to accept 40 dimensional visual inputs. The outputs of SOM can be regarded as the trajectory of a virtual robot that to predict and follow human actions.

The main components of MTRNN are called context layers, which are modeled by CTRNN. CTRNN is a special type of RNN which is a dynamical system model of biological neural networks. The output of each neuron in CTRNN is calculated not only by current input samples but also by the past history of neural states. This feature makes CTRNN possible for predicting continuous sensori-motor sequences [15].

The time constant τ of the two context layers is different. The time constant of fast context layer should be set smaller than slow context layer. Context units with smaller τ will change quickly as indicated by the following equation:

$$u_{i,t+1} = \left(1 - \frac{1}{\tau_i}\right) u_{i,t} + \frac{1}{\tau_i} \sum_{j \in N} w_{ij} x_{j,t} \tag{1}$$

where $u_{i,t}$ is the potential of *i*-th node at time step *t*. w_{ij} is the weight value between from *j*-th to ith unit and $x_{j,t}$ is the neural state of the *j*-th unit at time step *t*.

The activation rule of *i*-th unit at time step *t* is defined by the following formula:

$$y_{i,t} = \begin{cases} \dfrac{\exp(u_{i,t})}{\sum_{j \in Z} \exp(u_{j,t})} & if\ i \epsilon Z \\ f(u_{i,t}) & otherwise \end{cases} \tag{2}$$

$$x_{i,t+1} = \begin{cases} q_{i,t+1} & if\ i \epsilon Z \\ y_{i,t} & otherwise \end{cases} \tag{3}$$

where *f(x)* is a conventional sigmoid function.

The Back Propagation Through Time (BPTT) is used for training and update the weights between SOM nodes and context nodes.

3 Proposed Model

3.1 Motivation

After training, when human generates some action sequences, MTRNN is used to predict and follow human's action if the corresponding initial states are set. In order to employ MTRNN, a prior knowledge of the human actions is essential to assign the corresponding state. Then MTRNN is able to predict and follow human's action. However, if a wrong initial state is given, MTRNN cannot predict human action effectively.

In order to use MTRNN for human-computer interaction, it is crucial to properly recognize the class labels based on the current human action even if the wrong initial states are provided. In other words, the system should be penalized for an improper selection of the initial state if the selected initial state doesn't correspond to the human action during training. Moreover, if a continuous action sequence including multiple action sequences is shown, the model should recognize each action label in the order. In short, our work can be simply summarized as: how to recognize the current human action and then choose the correct action command to follow.

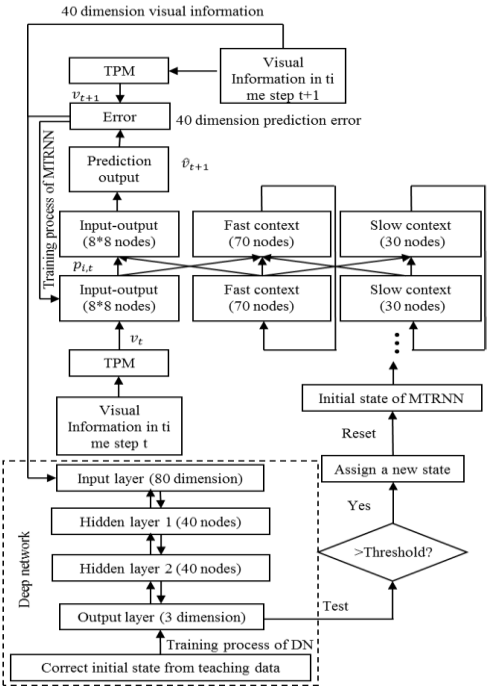


Fig. 2. Training process of the hybrid system

3.2 Computational Model

Fig. 2 shows the proposed computational model which hybrids between the conventional MTRNN for motion sequence estimation with the DN for deciding a suitable initial state for the MTRNN. MTRNN consists of two parts: fast context units and slow context units. 8*8 SOM is applied in input-output layer to receive the visual signal and output the prediction signal. KINECT, which is a depth based method, is used to collect the visual information. It is able to recognize 20 skeleton nodes in human body. The sensor of KINECT is capable of capturing data at 640x480 pixels in 30Hz. A 40 dimensional vector including the movement of 20 skeleton nodes (horizontal and vertical movement) is used as the visual input. Correspondingly, the output should also follow the 40 dimension for prediction.

30 nodes exist in slow context layer including 3 special nodes. These 3 special nodes are used for initial states according to the three different types of motion. Different action sequences should have different initial states.

We used a DN with EBP learning algorithm to make the correct decision for MTRNN. So the output of DN should correct initial states of the current sequences. The DN receives visual streams and decides the correct initial state. The training of the DN starts after the training of MTRNN is finished. So the feedback process of MTRNN is not executed during training process of the DN. The 40 dimensional visual information obtained using KINECT is used as a part of the visual input in the DN. But static posture recognition may not be enough to detect human action. Another part is obtained by the prediction error obtained from MTRNN. That means the input of DN in time step *t* consists of 40 dimensional visual information from KINECT in time step *t* and 40 dimensional data from MTRNN in time step *t-1* (totally 80 dimensional data are used). The output of DN is the corresponding initial states. Therefore, the hybrid system has two motivations: 1) to minimize the prediction error and 2) the object of DN is to find the correct initial states. The prediction error of MTRNN indicates the difference between consideration and measurement. By considering both prediction error and the current visual information, it is possible for the DN to find out which action is happening now.

If the output of DN gets the same result as the current initial states used in MTRNN, we do not need to reset the MTRNN. Thus we use a simple threshold rule to control whether the system need to change the initial states of MTRNN and initialize all the nodes value inside MTRNN. This rule is defined as:

```
Rule 1: If the output of action X is larger than 0.8
while the other outputs are smaller than 0.2 in three
successive times, then we can conclude that action X is
happening now.
Rule 2: If an action X does not correspond to the current
initial states of MTRNN, update it and reset all the
nodes value of MTRNN to the initial status.
```

We expect the DN to help the MTRNN to find the suitable initial state according to a situation. MTRNN will offer its prediction error to DN as an action response. DN and MTRNN should cooperate to meet a desired goal in each task.

After the DN gets the decision, it will be checked whether the current initial state (action state) in the MTRNN is correct or not based on the rules 1 and 2, and keep the state in MTRNN always correct.

4 Results

The experiments reported in this paper are performed on an IBM computer with Microsoft Windows 7 and an Intel Core (TM) 2 9400 2.67 GHz, 4GB Memory. We collected data corresponding to 3 different human actions: walking, running and

swing arms. Each group contains 10 samples. Each action is captured by KINECT within 3 seconds (90 frames). The main difference among the 3 kinds of action is that: the variation and speed in most skeleton nodes of walking is smaller than running; the hands trajectory of running and swing arms is similar but the legs are keeping static during swing arms.

20,000 iterations are applied to train MTRNN model. The mean square error (MSE), which is the average square error per neuron per step over all teaching sequence, converged to 0.001765. The process costs about 5.5 hours. After training of MTRNN model is finished, the training process of DN which also takes 20, 000 epochs costs about 22 minutes. The square error of DN per output unit per step converged to 0.004772.

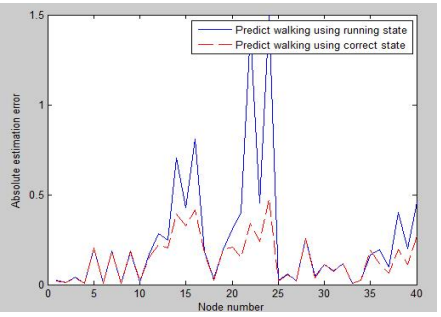


Fig. 3. Prediction result of the same action using different initial states

Table 1. Recognition result of 3 kinds of action

Human action	Recognition result %
Walking	91.06
Running	98.30
Swinging	98.70

Table 2. Recognition result of coninuous cases

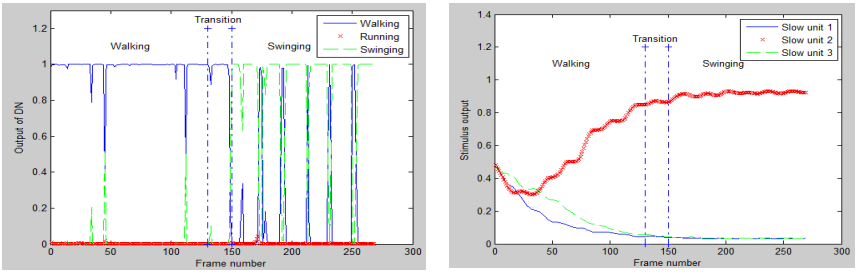
Human action	Recognition of First action (first 120 frames) %	Recognition of Second action (last 120 frames) %	Weighted average %
Walking+Swinging	91.21	94.17	92.69
Swinging +Walking	98.78	96.92	97.82
Walking+Running	92.17	99.77	95.97

Fig. 3 shows the situation of predicting same sequence using different initial state. The two error peaks corresponds to the movement of double hands. The prediction errors using wrong initial state predict larger error than correct one. The two peaks in Fig. 2 correspond to the skeleton nodes on the right and left hands. The movement of

double hands changed relatively more slowly than running. Thus the initial state corresponding to running cannot predict walking action well.

Table 1 shows the recognition result of a single action from DN. We can find that the recognition result is over than 91% each.

Our human is able to make a decision according to the circumstance. When human tester changes his action to a new one, our model should still be able to recognize and follow human action. We capture 3 kinds of continuous actions. The detail of these 3 sequences is shown in Table 2 Each sequence which lasts 9 seconds (270 frames) consists of 2 basic actions. It should be noted that these 3 sequences are not trained in the hybrid system before. All of the 3 sequences are started with the initial state corresponding to swing hands. The recognition results of Walking + Swinging is shown in Fig. 4 (a) and (b). The human changed his motion during the 120 ~ 150th frames. Most of the action sequences are successfully recognized in Fig. 4 (a). Some impulse noises also existed. Most of such impulse noises could be filtered out by using the rules 1 and 2 mentioned in Section 3.2. In [16], experiment results show that slow context units for initial states can be used for signal classification. For comparison, we record the status of 3 slow context units for initial states in these untrained continuous motion signals. However, no significant change can be found in Fig. 4 (b). That means conventional slow units based methods are not robust enough to detect motion changing.



(a) (b)

Fig. 4. (a) Prediction result of continuous signal (Walking + Swinging). (b) Conventional method based on slow units (Walking + Swinging).

5 Conclusion

In this paper, we proposed a hybrid network for human action recognizing and prediction. The method was evaluated by the experiments using KINECT. The prediction record from MTRNN is considered as the virtual trajectory of robot. The output of the DN is used to modify the initial state of MTRNN when the human changes his action. Experimental results show that the two kinds of network can work together. In addition to predicting the action correctly even when the initial state is wrong, the proposed hybrid network can also predict the changes in the human action. Finally, we can conclude that by having interactions between action and perception of the reality, it's able to perform the correct action according to the circumstance by comparing the difference between reality and idea.

Acknowledgements. This work was supported by the ISTDP (10044009) funded by MKE, Korea (70%) and also supported by the CRCP funded by MoEST (2012K001342) (30%).

References

1. Cruz, L., Djalma, L., Luiz, V.: Kinect and RGBD Images: Challenges and Applications. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, pp. 22–25 (2012)
2. Laptev, I., Lindeberg, T.: Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features. In: Kerckhove, M. (ed.) Scale-Space 2001. LNCS, vol. 2106, pp. 63–74. Springer, Heidelberg (2001)
3. Li, P., Bouzerdoum, A., Phung, S.L.: Adaptive Autoregressive Logarithmic Search for 3D Human Tracking. In: 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 18–21 (2012)
4. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976–990 (2010)
5. Wei, W., Yunxiao, A.: Vision-based human motion recognition: A Survey. In: 2009 Second International Conference on Intelligent Networks and Intelligent Systems, Tianjin, China, pp. 1–3 (2009)
6. Kendon, A.: Conducting Interaction: Patterns of behavior in focused encounters. Cambridge University Press, Cambridge (1990)
7. Joslin, C., El-Sawah, A., Chen, Q., Georganas, N.: Dynamic Gesture Recognition. In: Instrumentation and Measurement Technology Conference 2005, Ottawa, Canada, pp. 17–19 (2005)
8. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 93–202 (1980)
9. Hinton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
10. Behnke, S.: Hierarchical Neural Networks for Image Interpretation (2003)
11. Yamashita, Y., Tani, J.: Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. PLoS Computational Biology 4(11) (2008)
12. Cleeremans, A., Servan-Schreiber, D., McClelland, J.L.: Finite state automata and simple recurrent networks. Neural Computation 1(3), 372–381 (1989)
13. Hinoshita, W., Arie, H., Tani, J., Okuno, H.G., Ogata, T.: Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network. Neural Networks 24(4), 311–320 (2011)
14. Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43(1), 59–69 (1982)
15. Doya, K., Yoshizawa, S.: Adaptive neural oscillator using continuous-time back-propagation learning. Neural Networks 2, 375–386 (1989)
16. Zhang, Y., Ogata, T., Takahashi, T., Okuno, H.G.: Dynamic Recognition of Environmental Sounds with Recurrent Neural Network. In: The 28th Annual Conference of the Robotics Society of Japan, Nagoya, Japan (2010)

An Extended Version of the LVA-Index

Piotr Lasek

Institute of Computer Science, University of Rzeszów
ul. Prof. St. Pigonia 1, 35-310 Rzeszów, Poland
lasek@ur.edu.pl

Abstract. This paper presents the *ELVA-Index* - an extended version of the *LVA-Index*. It discusses its application as an underlying index structure for finding nearest neighbors in the *NBC* algorithm. The *ELVA-Index* combines the idea of approximation vectors from the *VA-File* and the layer approach. An important feature of the *ELVA-Index* is that for low-dimensional datasets, contrary to the *VA-File*, scanning the entire approximation file in order to find the nearest neighbors of a given query point is not required. Additionally, the *ELVA-Index*, in comparison to the *LVA-Index*, is faster and more efficient in terms of the time required to build it.

Keywords: clustering, index, nearest neighbors search, ELVA-Index, NBC.

1 Introduction

The problem of searching for nearest neighbors in datasets has been investigated over the past few decades. Over the years, a number of algorithms for searching data in multidimensional data spaces have been developed.

In 1974, Finkel and Bentley proposed the *Quadtree* [1], a solution which was easy to implement and efficient for two-dimensional spaces. The following year, Bentley proposed the *k-d tr*ee [2], which was also easy to implement and performed well when used for finding the nearest neighbors of a given point in low-dimensional spaces. Several years later, in 1981, Nievergelt and Hinterberger developed the *Gridfile* index [3]. The advantages of this index were its simple implementation and good performance. Unfortunately, the structure of *Gridfile* usually becomes unbalanced, which adversely affects the time searching in it. As Nievergelt et al. described in 1984 [4], *Voronoi diagrams* perform well only in low-dimensional spaces. In high-dimensional spaces, the computation of *Voronoi diagrams*, as well as the storage cost of the cells, is expensive. The retrieval times of nearest neighbors are not better than those in the methods described above. Other structures and algorithms developed over the years include the *R-Tree* index [5], which was similar to the *B-Tree* index [6] and is fast in low-dimensional spaces. However, because of the need to rebuild it often, basic operations, also very expensive. Additionally, a number of other indices were proposed later [7].

In 1997, the *Vector Approximation File* (*VA-File*) [8] was developed. Its characteristic feature is that it uses a mechanism of approximations by which a large number of irrelevant vectors can be easily excluded when searching through data.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 126–133, 2013.

Analyzing the *NBC* algorithm [9] we noticed that the *VA-File* was used as an underlying index structure for finding nearest neighbors. Further investigations led us to the idea of designing an *LVA-Index* [10] that employs the mechanism of approximations from the *VA-File* and the idea of layers from *NBC*. We examined the index by comparing it to the *VA-File* for datasets with a number of dimensions lower than 5. The results of the experiments showed that searching for nearest neighbors in the *LVA-Index* is faster than in the *VA-File*. In further research we improved our index and in this paper we will describe an extended version of the *LVA-Index*, namely the *ELVA-Index*.

This paper is divided into 6 sections. In Section 2, the *LVA-Index* will be presented again. In Section 3, the *ELVA-Index* - an extended version of the *LVA Index* - will be described. The experiments are discussed in Section 4. Section 5 gives a brief introduction to the *NBC* algorithm used to perform the experiments. Finally, in Section 6, conclusions are drawn and further works are proposed.

2 The *LVA-Index*

The *LVA-Index* is composed of so-called cells. Each non-empty cell representation contains a list of references to the set of points *P* belonging to it and a list of structures representing its nearest layers. Simultaneously, each point *p* representation stores the reference to a cell, to which *p* belongs.

This section presents definitions and notions related to the *LVA-Index*.

Definition 1. (*cell*) A *cell* c having coordinates $i_0, \dots, i_{d-1}$ is defined as follows:

$$c(i_0, \dots, i_{d-1}) = \{(x_0, \dots, x_{d-1}) : \forall j = 0 \dots d-1, a(j, (x_0, \dots, x_{d-1})) = i_j\}$$

where d is the number of dimensions, $j, i_j \in \mathbb{N} \cup \{0\}$ and $0 < j \leq d-1$. Function $a(j,p)$ is the approximation function [10] and its goal is to compute the value of approximation for a given point *p* in the *j*-th dimension. Assuming that the data space is divided into so called slices in all dimensions, the approximation of a point p in dimension j will be equal to the number of the slice, as shown in Figure 1. The *l*-th slice in the *j*-th dimension, where $j \in [0..d-1]$ and $j, l \in \mathbb{N} \cup \{0\}$, is denoted by $s(j,l)$.

Definition 2. (*layer*) The n-th *layer of a cell* $c(i_0, \dots, i_{d-1})$, denoted as $L_n(c(i_0, \dots, i_{d-1}))$, is defined recursively as follows:

$$L_n(c(i_0, \dots, i_{d-1})) = \bigcup_{\substack{s_0 = -n \dots n \\ \dots \\ s_{d-1} = -n \dots n}} \{c(i_0 + s_0, \dots, i_{d-1} + s_{d-1})\} \setminus \bigcup_{k=0..n-1} L_k(c(i_0, \dots, i_{d-1}))$$

Figure 1 shows a sample division of a data space into slices which are used to compute approximations of points. An example of how to compute such approximations is shown in Table 1.

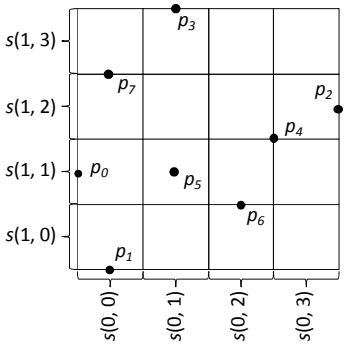


Fig. 1. A sample dataset of points divided into slices

Table 1. The results of computing the approximation function for points from the sample dataset from Figure 1

Point p	$a(0,p)$	$a(1,p)$
p_0	0	1
p_1	0	0
p_2	3	2
p_3	1	3
p_4	3	2
p_5	1	1
p_6	2	1
p_7	0	3

Example 1. Figures 2a and 2b illustrate two layers of cell $c(1,2)$: $L_0(c(1,2))$ and $L_1(c(1,2))$, respectively. $L_0(c(1,2))$ contains cell $c(1,2)$ itself and $L_1(c(1,2))$ contains the following cells: $c(0,1)$, $c(1,1)$, $c(2,1)$, $c(2,2)$, $c(2,3)$, $c(1,3)$, $c(0,3)$, $c(0,2)$.

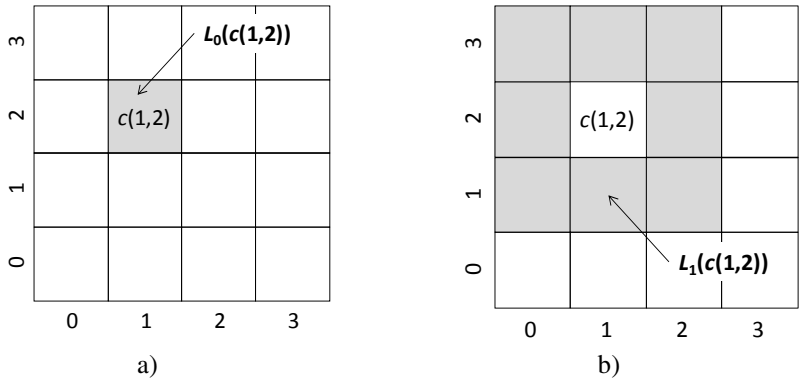


Fig. 2. Examples of layers defined in terms of cells

Example 2. Figure 3 gives an example of what the structure of the *LVA-Index* may look like for a sample two-dimensional dataset. In this case, the dataset contains points p_0, p_1, p_2, p_3, p_4, which belong to four non-empty cells: $c(0,3)$, $c(2,3)$, $c(0,1)$, $c(3,0)$. The *LVA-Index* for each of these cells stores a list of points P belonging to the cell and a list of the cell's nearest neighbor layers $L_i, i = 1,2$. 0-th layers are not stored in *LVA-Index*.

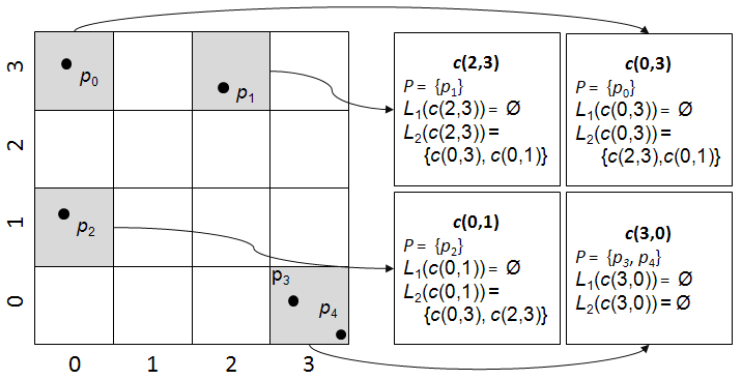


Fig. 3. An example of the *LVA-Index* for a sample two-dimensional dataset (number of dimensions $n = 2$, number of stored layers $l = 2$)

3 The *ELVA-Index*

In order to make the *LVA-Index* more flexible in terms of the number of stored layers, we propose the *ELVA-Index*, in which the number of neighbor layers stored in each cell is not fixed, but depends on the local densities of points in datasets. In other words, the number of layers stored in each cell is smaller if the neighborhood of the given cell is dense or greater if the neighborhood of the given cell is sparse.

In general, there is no difference in the structure of both the *LVA-Index* and *ELVA-Index* except that when building the latter, the number of layers to be stored is determined during the index building.

Below we present the *IterativeELVABuild* function which is used to build the *ELVA-Index*. This function was designed based on the *IterativeLVABuild* function for building the *LVA-Index*. The *IterativeELVABuild* function, when building the index, iterates until the number of neighbors found (*neighborsCount*) is less than k and the set of cells belonging to the l_i-th layer is not empty. This implies that the number of iterations can vary depending on the density of a dataset.

function *IterativeELVABuild*(*D, LVA, k*)

```
1. /* k     – the layer number
2.    d     – number of dimensions
3.    D     – dataset */
4. forall p ∈ D do
5.    cell = LVA.DetermineCell(p);
6.    AssignVector(cell, p);
7.    neighborsCount = 0;
8.    li = 0   // layer index
9.    while (neighborsCount < k and L_li ≠ ∅) do
10.       L_li = IterativeGetLayerCells(LVA, c, li)
```

```
11.       forall c ∈ L_li do
12.           neighborsCount += |c.points|;
13.           R_li= IterativeGetLayerCells (LVA, c, li);
14.           if (cell ∉ R_li) then
15.               R_li. AddNonEmptyCell(cell)
16.           endif
17.       endfor
18.     li = li + 1
19.   endwhile
20. endfor;
```

Example 3. The difference between the *LVA-Index* and *ELVA-Index* with respect to the number of scanned nearest layers is illustrated in Figure 4. It can be seen that the number of nearest layers to be scanned is fixed and equal to 2 in the case of the *LVA-Index* and the number of nearest layers is determined dynamically in the case of the *ELVA-Index*.

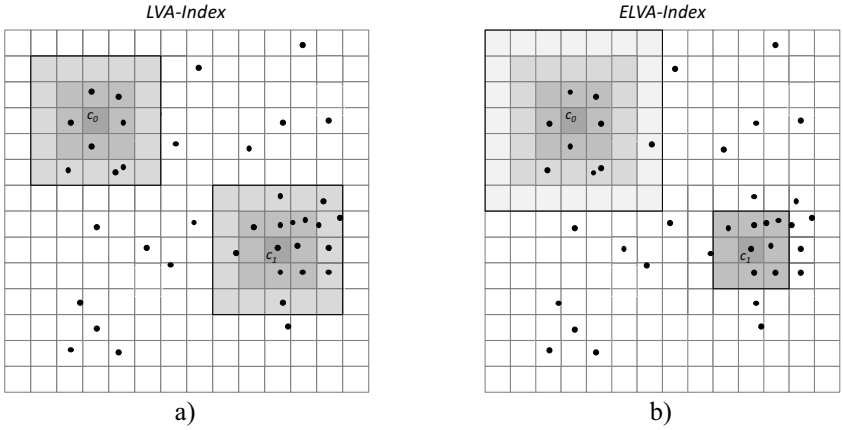


Fig. 4. The difference in building the *LVA-Index* (a) and *ELVA-Index* (b)

Searching in both the *LVA-Index* and *ELVA-Index* can be performed using the same *ELVA-Index Simple Search Algorithm* (*ELSSA*). The function is designed so that the *InitCandidate* function initializes an array of length k which will contain the neighbor points after the search has finished and the *Candidate* function checks if a currently-processed point p is really a candidate to be one of k nearest neighbors of point q. If p is a candidate, then the *Candidate* function updates the *neighbors* list.

```
function ELSSA(D, q, k)
1.  /*  D          – dataset of points
2.      q          – query point for which the neighbors are to be found
3.      k          – number of neighbors to be found
4.      neighbors –list of neighbors to be determined   */
5.
6.  max = InitCandidates(neighbors, k, ∞); // the radius is set to ∞, so the area of
7.  searching is not limited
```

```
 8. foreach layer in q.cell.L do                    // the main loop of LSSA
 9.     foreach cell in layer do
10.         p = first point in cell
11.         if (lb(p, q) < max) then
12.             foreach r in cell do                    // r is a candidate point
13.                 δ = dist(p, q)
14.                 max = Candidate(p, δ, neighbors)
15.             endfor
16.         endif
17.     endfor
18. endfor
19. return neighbors having coordinates different from ∞;
```

4 The *NBC* Algorithm

The *Neighborhood-Based Clustering* (*NBC*) algorithm is a density-based clustering algorithm. Its characteristic feature is the ability to measure relative local densities as well as the ability to discover clusters of different local densities and of arbitrary shape. It takes only one input parameter (a k value). The authors of *NBC* proposed using a cell-based approach to calculating kNB by cutting the data space into (high-dimensional) cells and using the *VA-File* to organize the cells. We have used this algorithm to test how the *ELVA-Index* performs when used as an underlying index for the clustering algorithm.

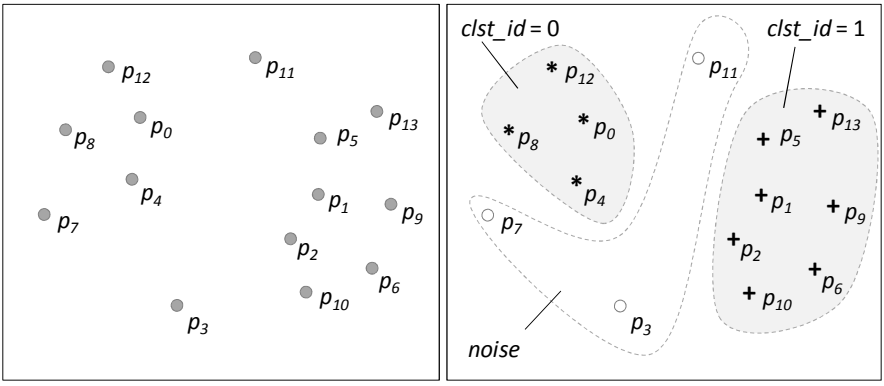


Fig. 5. A sample dataset and the result of clustering using *NBC*

The *NBC* algorithm starts with the *CalcNDF* function which calculates the NDF factor for each point p_i in dataset D ($i = 0$ ... $|D|$). After calculating the NDF factors for each point in *D*, the clustering process is performed. For each point p_i, it is checked if $p_i.ndf$ is less than 1. If $p_i.ndf < 1$ then, at this moment, p_i is omitted and a next point p_{i+1} is checked. If $p_{i+1}.ndf \geq 1$, then, p_{i+1} as a local core point is assigned to the currently-created cluster identified by the current value of *ClusterId*. Next, the *DPSet* variable for storing points, which are directly neighborhood-based density-reachable from point p_{i+1}, is cleared and each point, say q_j, belonging to $k^+NN(p_{i+1}^-)$ (a punctured neighborhood of point p_{i+1}) is assigned to the

currently-created cluster. Moreover, if $q.ndf$ is greater than or equal to 1, then point q is added to *DPSet*. Next, *DPSet* is analyzed. For each point from *DPSet*, say r_m, it is checked if the point has already been assigned to a cluster. If it has, then the next point r_{m+1} is processed. If the point has not been assigned to any cluster, then it is assigned to an appropriate cluster ($r_{m+1}.ClusterId = ClusterId$) and if $r_{m+1}.ndf \geq 1$, then it is also added to *DPSet*. Finally, all unclassified points are marked as noise by setting the value of their *ClusterId* attribute to *NOISE*.

Figure 5 presents a sample non-clustered input dataset and the result of clustering using the *NBC* algorithm.

5 Experiments

In order to examine the efficiency of the *ELVA-Index* we have performed a series of experiments. Prior to that, we implemented three versions of the *NBC* with different underlying indices, namely: the *VA-File*, the *LVA-Index* and the *ELVA-Index*. These indices take different parameters, so before running the tests we experimentally determined the most appropriate values of the parameters and used them in all tests. In total, we performed 21 tests. We used three datasets having 2 and 3 numerical attributes and containing up to 50000 data objects.

Table 2. A comparison of runtimes of *NBC-VAFILE*, *NBC-LVA* and *NBC-ELVA* – the implementations of *NBC* with different underlying indices

No	Dataset	Runtimes [ms]					
		NBC-VAFILE $b = 5, k = 10, l = 2$		*NBC-LVA* $b = 7, k = 10, l = 7$		*NBC-ELVA* $b = 7, k = 10$	
		ind.	clust.	ind.	clust.	ind.	clust.
1	sequoia_2000_d2_2503	32	678	532	796	156	437
2	sequoia_2000_d2_3910	47	19907	875	2890	250	1281
3	sequoia_2000_d2_5213	62	37360	1219	5422	375	3047
4	sequoia_2000_d2_6256	93	62110	1531	8641	469	5484
5	manual_d2_2658	31	10500	766	984	234	578
6	manual_d2_14453	78	607797	3641	49359	174	25984
7	random_d3_50000	15375	1458969	265281	6188	107297	2750

The results obtained are given in Table 2 and prove that for the datasets used in the experiment, the *ELVA-Index* performs better than the other indices. In some cases, the implementation of *NBC* with the *ELVA-Index* outperforms the other implementations by even up to two orders of magnitude.

6 Conclusions and Further Works

The proposed *ELVA-Index* as well as the methods for building it and using it to search for nearest neighbors were presented in this paper. The characteristic feature of the *ELVA-Index* is that it employs the idea of approximation vectors and the layer approach for determining nearest neighbors. Additionally, the number of layers stored in the index is not constant but depends on the local densities of points in an analyzed dataset. The experiments performed show that the efficiency of the *NBC* algorithm with the *ELVA-Index* as the underlying structure for searching for nearest neighbors is better than the implementation of *NBC* with the *VA-File* or *LVA-Index*.

Planned future research will focus on the quality of clustering algorithms using the *LVA* and *ELVA* indices, as well as on employing the idea of the Triangle Inequality Property for improving the efficiency of searching for nearest neighbors as described in [11].

References

1. Finkel, R., Bentley, J.: Quad-trees: A data structure for retrieval on composite keys. ACTA Informatica 4(1), 1–9 (1974)
2. Bentley, J.: Multidimensional binary search trees used for associative searching. Communications of the ACM 18, 509–517 (1975)
3. Nievergelt, J., Hinterberger, H., Sevcik, K.: The grid file: An adaptable symmetric multikey file structure. ACM Transactions on Database Systems, 38–71 (1984)
4. Aurenhammer, F.: Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23(3), 345–405 (1991)
5. Guttman, A.: R-Trees: A dynamic index structure for spatial searching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Boston, MA, pp. 47–57 (1984)
6. Bayer, R.: Binary B-trees for virtual memory. In: Proc. ACM-SIGFIDET Workshop on Data Description, Access and Control, pp. 219–235 (1971)
7. Böhm, C., Berchtold, S., Kriegel, H.-P., Michel, U.: Multidimensional Index Structures in Relational Databases. J. Intell. Inf. Syst. 15(1), 51–70 (2000); Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann Publishers (2000)
8. Weber, R., Schek, H.-J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of the 24th VLDB Conference on Very Large Data Bases(VLDB 1998), New York City, NY, pp. 194–205 (1998)
9. Zhou, S., Zhao, Y., Guan, J., Huang, J.: A neighborhood-based clustering algorithm. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 361–371. Springer, Heidelberg (2005)
10. Lasek, P.: LVA-Index. An Efficient Way to Determine Nearest Neighbors. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 623–633. Springer, Heidelberg (2009)
11. Kryszkiewicz, M., Lasek, P.: A neighborhood-based clustering by means of the triangle inequality. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 284–291. Springer, Heidelberg (2010)

Anomaly Monitoring Framework Based on Intelligent Data Analysis

Prapa Rattadilok[1], Andrei Petrovski[1], and Sergei Petrovski[2]

[1] School of Computing Science and Digital Media, Robert Gordon University, UK
{p.rattadilok,a.petrovski}@rgu.ac.uk
2 School of Electric Stations, Samara State Technical University, Russia
petrovski@rambler.ru

Abstract. Real-time data processing has become an increasingly important challenge as the need for faster analysis of big data widely manifests itself. In this research, several Computational Intelligence methods have been applied for identifying possible anomalies in two real world sensor-based datasets. By achieving similar results to those of well respected methods, the proposed framework shows a promising potential for anomaly detection and its lightweight, real-time features make it applicable to a range of in-situ data analysis scenarios.

Keywords: intelligent data analysis, automated fault detection, big data, real-time, K-Means.

1 Introduction

Due to the fast pace of computing technology advancements in terms of both memory capacity and processing speed, increasing amounts of high precision data is becoming available. With this vast amount of data coming on stream, traditional data acquisition and data processing methods have become inefficient or sometimes inappropriate. Heterogeneous data sources also add complexity in the form of analytical challenges, especially when there exists time and/or cost differences in processing data from different sources. It has become more important than ever before (especially in a real time environment, where the processed information is only useful until a particular point in time and excessive latency would render the information useless [1]) to be able to effectively distil the large amount of data into meaningful information, as well as optimally selecting the data sources to minimise the time and/or cost.

The detection of outliers (or anomaly) is the process of finding patterns in a given data set that do not conform to an expectance and is the subject of much recent research [8]. This can be especially important in the case of big data where data volumes and sample rates limit the amount of data that can be simultaneously processed. Machine learning techniques are commonly used, but rely heavily on human knowledge integration as well as human involvement in defining every possible anomalous pattern, which are often unknown *a priori*. The resulting algorithms tend to be very

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 134–141, 2013.

problem-specific, and in most cases they are even designed specifically to evaluate pre-defined types of input, which makes them inappropriate in dynamically changing environment or in new problem domains.

2 Automated Anomaly Detection

Considering the example of automotive process control, various sensors may be used for different types of analysis. An anomaly in the air conditioning system, whilst the fan sensor indicates fully functioning blades, could imply a blockage, which requires manual/visual inspection inside the ventilation holes. Activating sensors in an ad-hoc basis and timely manner can help to reduce processing cost, as well as the amount of data available for analysis. Ideally, the data that is obtained from simple sensors should be chosen for anomaly detection. More detail-rich data from sophisticated sensors can later be activated, the resultant data undergoing further investigations. Fig. 1 illustrates the system overview which exploits both fast and detail-rich data.

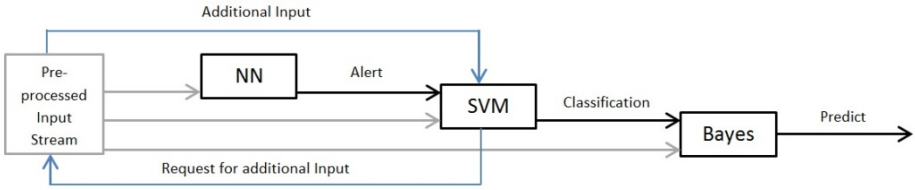


Fig. 1. System overview

The three Computational Intelligence (CI) techniques shown in Fig. 1, which are responsible for the identification, classification and prediction of anomalies, are Artificial Neural Network (ANN), Support Vector Machine (SVM) and Bayesian network (Bayes) respectively. This framework has been successfully applied [2], where the ANN operates on a fast data input stream. Experimental conditions (ie. learning rates, momentum and number of hidden units) were auto-configured using the standard WEKA configuration framework. Depending on the characteristics of the identified anomalies, either additional fast data may be required or detail-rich data may be processed to improve the accuracy of anomaly classification using an SVM, whereas the probabilities of future anomalies can be estimated using a Bayesian network. The abovementioned techniques are compared in [3].

3 Anomaly Identification Framework

To define properties and/or patterns of anomalies in advance is not a trivial task, especially within novel problem domains. It is also possible that what once was anomalous may become acceptable in a dynamic environment. One possible solution to pre-defining properties and/or patterns of anomalies within novel problem domains and/or dynamic environment is by combining the use of mathematical deviation analysis and computa-

tional intelligence techniques to evaluate the data. The mathematical deviation analysis allows the identification of undefined anomalies by evaluating the deviation from expectation, whereas computational intelligence techniques identify anomalies based on pre-defined patterns or provide continuous learning while solving the problem. Fig. 2 illustrates the anomaly identification process for the first sub-process (labelled NN) of the three-stage process in the system overview shown in Fig. 1.

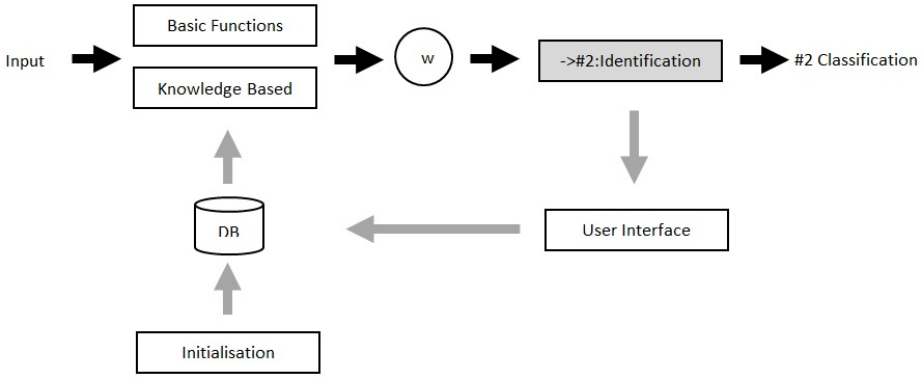


Fig. 2. Anomaly identification process

The "Knowledge-Based" process identifies an anomaly based on pre-defined patterns. Also, this process adaptively selects a suitable computational intelligence technique together with required parameters for the current problem solving stage. Any pre-defined patterns can be specified via the "Initialisation" process. The "Basic Functions" process mathematically evaluates the incoming input data in order to identify anomalies based on their deviation from the expectations.

Non-numerical input streams from sensors can all be manipulated into some form of numerical data. For example, images in video sequence can be evaluated in the sense of colour deviations or colour clustering, object trajectory, or even the intermittency of some input streams. The deviation analysis of numerical data allows for a more general approach to identifying anomalies, especially when the features of the anomalies are not known a priori.

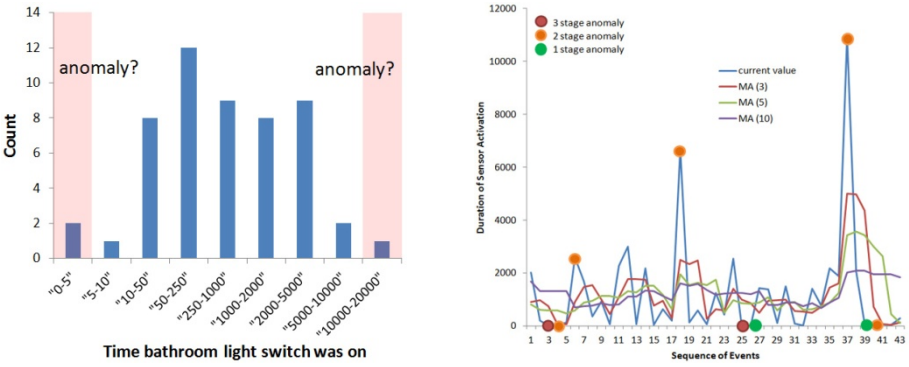
The results from both the "Basic Functions" and "Knowledge Based" processes are combined based on the degree of belief ("w" in Fig. 2) that varies over time depending on how accurate the "Knowledge Based" process is at identifying the anomalies. The "User Interface" process allows the user to provide anomaly confirmation/rejection responses. Only fast and/or explicit data will be used as the input for this anomaly detection process. The identified anomalies, as an output of the identification process, are passed onto the classification process, where association of anomalies with meaningful inferences can eventually be used to predict future process states. Additional inputs might be acquired at later stages to enhance accuracy.

4 Experimental Setup and Results

Two datasets from two different fields are used in the experiments: a multi-sensor smart home environment and an automotive process control. The application programs are developed using Java with the Encog library [4]. The mathematical deviation analysis for the "Basic Functions" process is evaluated using three different window lengths: short-, medium- and long-term. A number of computational intelligence techniques This included an ANN approach using multi-layer perceptron (MLP) with back propagation of error (back-prop) and an SVM approach using a sequential minimal optimization algorithm for training a support vector classifier; all experimental setup parameters are determined empirically.

4.1 Multi-sensors Smart Home Environment

The research of smart home environments has grown in recent years due to the aging population over the world and an increase in availability of inexpensive sensors. The motivation of the research is to ensure a safe living environment and lifestyle for smart-home occupants by monitoring the occupants' movement, behaviour and interactions with objects/appliances on a frequent basis. The dataset used in this paper was obtained by non-invasive monitoring of object and appliance manipulation [5]. Changes in activity patterns, deviations in terms of regularity or duration of different activities are examples of possible anomalies, but without ground truth it is impossible to know for sure what should be classed as anomalies. Fig. 3a highlights possible anomalies at high and low values of the available sensors based on the duration, as well as regularity, of interactions. Fig. 3b highlights the anomalies detected using the "Basic functions" process based on shorter-, medium, and longer-term memories of 3, 5 and 10 recent events respectively. The sensor output is considered anomalous if the current value is three times bigger or fifty times smaller than expected.



(a) **(b)**

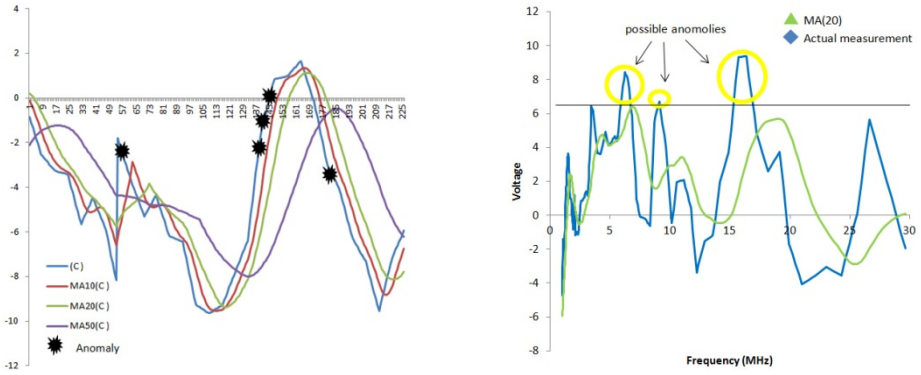
Fig. 3. Possible anomalies within the smart home dataset

Considering two examples: 1) at 9pm, on a Monday, the bathroom light was switched on for 5 seconds, and only bathroom cupboard sensors was activated (i.e. open and close); 2) at 11am, on a Monday, the bathroom light was switched on for 20,000 seconds, no other sensors are activated.

Information from other sensors and/or the context based information (e.g. time of day or day of week) can help with the identification process. The idea is to quickly identify possible anomalies based on the deviation analysis, and to exploit further the data related to the identified anomalies, if required, at the later stages. Fig. 3b illustrates that the best coverage of anomalies is obtained using the deviations based on the results of two out of three stages. When only one or all of the stages are used, the number of detected anomalies deteriorates. As previously stated, the anomaly confirmation/rejection depends on the user or expert knowledge. The "Knowledge Based" process is trained based on the anomaly confirmation-rejection responses from the user. The neural network accepts two inputs (i.e. current values and deviations) and produces the anomalous identification as an output. It is possible to achieve the accuracy rate of about 78% on an unseen test set using 5-fold cross-validation.

4.2 Automotive Process Control

The next dataset is based on the effect of an interference-suppression capacitor in terms of noise at different frequencies when the capacitor is connected to the bonding or to the engine cylinders. Anomalies in interference voltage can be detected in any of the three options of connecting the interference-suppression capacitor when the noise level is changing too abruptly (Fig. 4a) or significantly exceeds a threshold (Fig. 4b). The same "Basic Functions" process as applied to the multi-sensor smart home environment dataset is adopted in these experiments.



(a) (b)

Fig. 4. Possible anomalous interference voltage

In Fig. 4a, the small, medium and large diapasons of frequency values are 10, 20 and 50 MHz respectively. Instead of using a single value, the intervals of ±1.25, ±2.5 and ±7.5 dBμV are used to compare between the actual value and the mathematical

deviation. If the differences are larger or smaller than the specified limits in at least two of the three ranges of frequencies, then the value is considered anomalous. In Fig. 4b, anomalies are determined by comparing the actual values to the highest peak obtained using the deviation analysis.

Fig. 4a is derived using the data on the interference voltage for frequencies above 65 MHz when interference-suppression capacitor connected to the engine is used. As shown, the anomalous frequency diapasons are 53-56, 133-150 and 177-180, this equates to the frequency ranges of 86.97-87.30, 96.25-98.34 and 101.76-102.15 MHz. Fig. 4b is derived using the data on the interference voltage for frequencies below 30 MHz when no interference-suppression capacitor is used. As can be seen from the figure, three anomalies are identified. The result coincides with the expert knowledge obtained related to excessive noise as discussed in Fig. 5a and 5b.

In the dataset used, the threshold values for acceptable noise are not specified for all the. Fig. 5a and 5b demonstrate the results of applying two computational intelligence techniques to estimating the missing thresholds. Fig. 5a compares the actual threshold and the predicted threshold using the "Knowledge Based" process together with a linear regression technique (Eq. 1), where the variables $V_i, i = \overline{1,4}$ are frequency, interference voltage with suppression capacitor connected to the bonding, interference voltage with no suppression capacitor, and interference voltage with suppression capacitor connected to the engine cylinders respectively.

$$T = (0.0825 * V_1) + (0.0634 * V_2) + (-0.1852 * V_3) + (0.3486 * V_4) + 9.7531 \quad (1)$$

Fig. 5b represents the thresholds of 6, 9 and 15 as semantic classification targets rather than linear values. The value of 47.3 is the mid-point between the lowest value for any threshold 15 sample (65) and the highest value for any threshold 6 sample (29.6). The value of 0.416 is the mid-point between the lowest value for any threshold 6 sample (0.5333) and the highest value for any threshold 9 sample (0.2982).

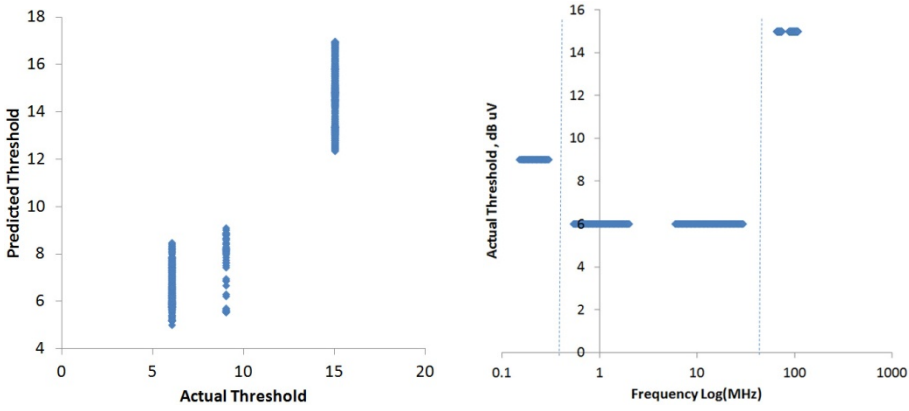


Fig. 5. Using computational intelligence techniques to determine interference voltage threshold

From Fig. 5a, it is apparent that samples with a threshold of 15 are accurately predicted, but there are some (60) mis-classifications of samples with a threshold of 6 and 9. This suggests some non-linearity in the data that is difficult to predict using linear regression. For 'Big' data this initial approach of building a linear regression model is useful because it enables assessing difficulty and non-linearity of the relationships within the dataset. A non-linear regression approach may be more accurate. Fig. 5b confirms the finding when frequency is plotted against the actual threshold. The clear segregation of the three thresholds based of frequency is apparent. There also appears to be at least 2 sub populations for the samples with the threshold of 6 and 15.

4.3 K-Means Comparisons

Due to the lack of ground truth about anomalies in the datasets, algorithmic validation and performance comparison are achieved through assessing the level of agreement with other methods. Clustering is a well-established method for detecting outliers and anomalies in a dataset [6]; a constrained k-means version [7] of the clustering algorithm is applied to the data. The results of the clustering approach on the smart home environment data and automotive process control data are shown in Fig. 6a and 6b respectively. Similar results are achieved by clustering as by the proposed approach.

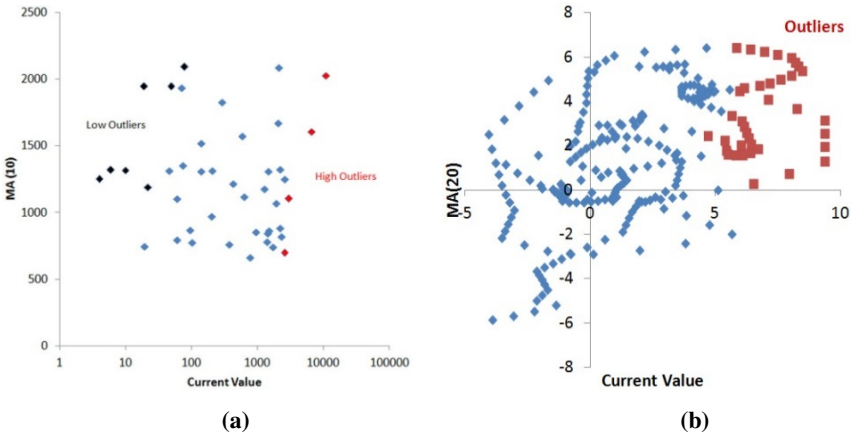


Fig. 6. Comparisons with K-means clustering

In Fig. 6a, by specifying three clusters, high and low outlier regions to the core data can be generated. These two outlier clusters contained all the values highlighted in Fig. 3b plus one extra high outlier and one extra low outlier. In Fig. 6b, two clusters are specified, in order to identify the values above and below the threshold. The outlier clusters contained all the values highlighted in Fig. 4b. So while the results of the two methods are similar, the proposed approach has several advantages that include the ability to detect anomalies in real time without the need to carry out clustering on the whole dataset and the comparatively lower computation load. Both of these advantages are especially important for real time monitoring scenarios.

5 Conclusions

Activating various sensors and processing their data incurs different costs. With the large volumes of data from heterogeneous sensors, evaluating explicit sensor data in order to identify an anomaly before exploiting detail rich and/or more expensive sensor data can be beneficial, especially when (near) real-time decisions are required.

In many cases, a specific problem domain can be unexplored, and the generality of the approach can boost the applicability of the system with little or no modification. The deviation analysis is combined with computational intelligence techniques to tackle both explored and unexplored problem domains. The approach has been successfully applied to two datasets from very different fields of knowledge.

When comparing the proposed approach with a K-means clustering technique, comparable results are obtained. This supports the validity of the approach in identifying anomalies by using the proposed deviation analysis. Furthermore, the approach is generic enough to tackle different datasets from two very different problem domains.

Currently, all parameters are determined empirically both in the "Basic Functions", as well as the "Knowledge Based", processes. Various sensors may require different settings, and these values may vary over time. The acceptable range of deviation may change over time and between various sensors. Automatically determining optimal parameters, as well as adaptively adjusting them in real time, would improve the versatility of the algorithm with little or no modifications. This will be investigated further, and the identification output will be added to the next two sub-processes, classification and prediction, providing better insight into the nature of various anomalies.

References

1. Jacobs, A.: The Pathologies of Big Data. ACMQueue (2009)
2. Rattadilok, P., Petrovski, A.: Inferential Measurements for Situation Awareness: Enhancing traffic Surveillance by Machine Learning. In: CIVEMSA 2013 (to appear, 2013)
3. Correa, M., Bielza, C., Pamies-Teixeira, J.: Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process. Expert Systems with Applications 36(3), 7270–7279 (2009)
4. http://www.heatonresearch.com/encog
5. http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/
6. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22(2), 85–126 (2004)
7. Tung, A.K.H., Han, J., Lakshmanan, L.V.S., Ng, R.T.: Constraint-Based Clustering in Large Databases. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 405–419. Springer, Heidelberg (2000)
8. Dereszynski, E.W., Dietterich, T.G.: Probabilistic models for anomaly detection in remote sensor data streams. In: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence. arXiv:1206.5250 (2012)

Customer Unification in E-Commerce

Marcin Gorawski[1,2], Aleksander Chrószcz[1], and Anna Gorawska[1]

[1] Silesian University of Technology,
Institute of Computer Science,
Akademicka 16, 44-100 Gliwice Poland
{Marcin.Gorawski,Aleksander.Chroszcz,Anna.Gorawska}@polsl.pl
[2] Wroclaw University of Technology,
Institute of Computer Science,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Marcin.Gorawski@pwr.wroc.pl

Abstract. Data mining applied to social media is gaining popularity. It is worth noticing that most e-commerce services also cause the formation of small communities not only services oriented toward socializing people. The results of their analysis are easier to implement. Besides, we can expect a better perception of the business by its own users, therefore the analysis of their behavior is justified. In the paper we introduce an algorithm which identifies particular customers among not logged or not registered users of a given e-commerce service. The identification of a customer is based on data that was given so as to accomplish selling procedure. Customers rarely use exactly the same identification data each time. In consequence, it is possible to check if customers create a group of unrelated individuals or if there are symptoms of social behavior.

1 Introduction

When we posses a complete database of selling transactions then we can apply a group of well known algorithms[58] such as the analysis of shopping basket contents or the analysis of the most frequently sold items. Problems emerge when the volume of customers and transactions is small. Then, there is a huge risk that the current data excludes some groups of people which contain potential new customers. The weakness of the above methods manifests itself also in the case of frequently changing ranges of goods. This happens on the pharmaceuticals & cosmetics market which rapidly implements scientific discoveries. As a result, historical sales analysis has little usefulness when new products are considered.

In this area, multi dimensional data mining of social networks[61], news portals[2] and auction portals has become a promising solution in contrast data mining of streams [14,13,40,44,51,50,39,12,46,45,20] and spatial data [5,8,42,43,30,55,31,29,22,7,9,26,6,49,47,52,18,32,19,28]. Thanks to a huge volume of users registered in such services, those systems are considered to be an unbiased opinion centers. There are projects extracting hot topics from Twitter messages by means of text analysis[61]. Other projects are designed to trace

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 142–152, 2013.

knowledge propagation[1] in Internet communities. As a result, we can gain access to opinions about new products and advertising campaigns.

Social analysis are valuable also for smaller e-commerce systems. The paper addresses the problem of customer identification in selling transactions collected from some e-shopping system. The aim is to recreate customer list and assign transactions to them. The incomplete attribute set constitutes the main challenge. Customers which are not registered leave a lot of attributes empty. It was popular that they fill in e-mails or telephone numbers exclusively. Besides it is possible that customers use data which does not identify them. For instance the shipping address may not be the customer's own address. In order to solve the above task, we have designed a clustering algorithm which groups together transactions committed by the same customer. In contrast to a typical clustering task, this solution is expected to generate thousands of clusters. In consequence, there is required the algorithm which automatically estimates the cluster number. Moreover, the distance function applied to compare the similarity between clusters should work even some attribute values are missing. Those specific requirements inspired us to create a new algorithm. Among potential strategies [33,36,62,4] we have chosen agglomerative one because this algorithm can be adapted to work with our custom function which measures distance between clusters.

This problem appeared in a real system. We have an opportunity to carry out data cleaning and data integration for a medium sized e-commerce service existing for a few years. In order to attract reluctant customers, this e-commerce systems do not oblige all customers to create their account. Particularly infrequent or sporadic users do not create accounts. On the other hand, knowing their profiles would enable that company to acquire them as a regular customers.

The paper is organized as follows: Section 2 introduces the data structure of processed data. In Section 3 our clustering algorithm is defined. Next, Section 4 contains our experiments. Finally, in Section 5 we conclude our results.

2 Data Structure

Let us assume that there are two tables: $User$ and $Transaction$ depicted in fig. 1. The $Transaction$ table defines which user in a given day has bought goods. Each transaction is accompanied by attributes describing billing and shipping address details like: post code, country, state and address. In order to make the model simple, the list of products in transactions is omitted. Because the customer's address, e-mail and identification attributes may change in time for a given customer, the valid values are stored separately in the *transaction* table when a transaction is submitted.

In order to check if two transactions were concluded by the same customer it is necessary to compare the transaction attributes. As we can see, the data scheme allows us to store information about transactions on low granularity level. On the other hand, only a narrow group of attributes are mandatory. As a result, a considerable number of null values exist in the database. Especially, billing

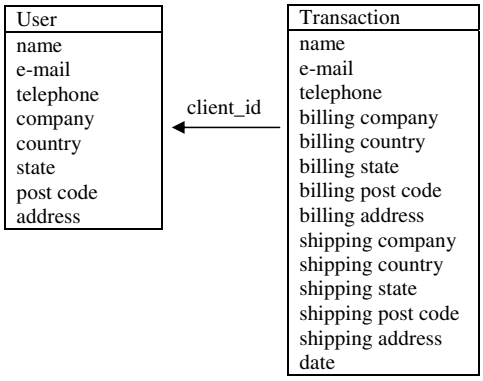


Fig. 1. Schema of data source

addresses details were rarely filled in the datasets on which we have worked. Bellow we describe other data imperfections which were present in the processed data.

It may seam that the *name* and *company* name will be helpful in the identification of a particular customer. In reality, customers are usually not consistent and use plenty of abbreviations when the name is long.

In contrast to previous attributes, e-mail values were always verified in the system. Despite this fact, particular customers cannot be identified by means of this attribute only for two reasons: customers usually have a few e-mails; besides, a group of transactions are concluded on behalf of somebody. This occurs when office work is delegated to subordinates. Such information may be strategic, therefore, we will also consider this aspect in the paper.

Another noteworthy thing is that a telephone number combines a few pieces of information. A complete telephone number is useful when it is a personal telephone number. On the other hand, we can analyze only the prefix. In consequence, we can identify with high probability people coming from the same company.

The limitations of address usage are similar to those diagnosed for *company* name and customer *name* attributes. Therefore, it is important to unify those values before comparison. It was common in the analyzed data that a customer sometimes inserts unnecessarily a room or suite number, dots, comas and empty spaces. In order to make the address unification process more resistant to possible textual variations, we have prepared a dictionary of popular abbreviations. In consequence, we can arrive at an address with all the possible abbreviations applied regardless of which address version the algorithm started with.

3 Algorithm

Our clustering algorithm is a combination of agglomerative clustering [60,3,64] with a distance function based on the Naive Bayesian classifier[59]. The other so-

lutions like k-means and canopy [62] clustering are better suited for problems with fewer clusters. There are series of indexes developed by us which lowers calculation complexity [14,40,30,55,48,27,54,38,25,17,24,37,53,16,21,10,34,56,35,23,11,41]. Because explored real dataset contains relatively small dataset, we have decided to use agglomerative clustering, because the resulting clusters are easier to describe by means of reports required by company personnel.

Our clustering process is divided into two phases. During the first phase, the Naive Bayesian classifier is taught. It checks if two transactions were concluded by the same customer. The answer is expressed as:

$$similarity = pg/(pg + npg) \tag{1}$$

where:

- pg - the probability that two clusters of transactions were concluded by the same customer,
- png - the probability that two clusters of transactions were concluded by different customers.

When *similarity* is grater than 0.5 it means that both transactions belong to the same customer. Values pg and png approximate the probabilities defined above. Therefore it is necessary to use the *similarity* value which normalizes pg according to $pg + npg$. The algorithm of pg and png estimation will be explained in subsection 3.2.

The next phase implements the agglomerative algorithm which combines the most similar transactions until there are no clusters with similarity greater than 0.5.

3.1 Agglomerative Clustering

Agglomerative clustering [63] is a greedy algorithm which implements a bottom-up strategy. Each observation starts in its own cluster, then according to a distance function the closest pairs of clusters are merged into a single larger cluster. The process repeats until a single cluster is created. The customizations of this algorithm are used in a wide range of applications and fields[60,4,57,64] including data mining, compression and image processing.

In order to optimize the basic algorithm version, we have introduced a heap which preserves information on the similarity of transactions [63]. At the beginning, the heap is initialized with pairs of the most similar clusters. Each cluster has the most similar one assigned to it. When new cluster C is created from clusters A and B then pairs containing either A or B are no longer valid in the heap. Those entries can be immediately removed. The introduced algorithm removes those elements lazily. Each time another element from the heap is processed, it is checked if any cluster from the pair has already been incorporated into a bigger cluster and removed from the *clusters* list. When this is true, the further analysis of this element is omitted. The lazy strategy simplifies the algorithm

and makes it faster. The detailed description of this algorithm can be found in [63].

3.2 Distance Function

We have decided to measure the similarity between transactions by means of the Naive Bayesian classifier because it allows us to compare attributes with missing values. Let us assume that we have two transactions to be classified. At the beginning, the corresponding attributes of those transactions are compared and the result is stored in new record R. The comparison of two corresponding attributes may yield one of the three situations: equal, not equal and unknown. The last outcome occurs when at least one transaction has no defined value for the compared attributes. According to values stored in R, the Naive Bayesian classifier calculates the probability of concluding transactions by the same customer.

This classifier is based on the bellow probability model:

$$p(C|F_1, ..., F_n) = \frac{p(C)p(F_1, ..., F_n|C}{p(F_1, ..., F_n)} \tag{2}$$

The variable C has two states which indicate whether two transactions were concluded by the same customer or not. Variables $F_1, ..., F_n$ represent similarity of transaction attributes. This model leads to the following Naive Bayesian classifier rule:

$$\begin{array}{l} classify(f_1, ..., f_n) = \\ \underset{c}{\operatorname{argmax}}\, p(C = c) \prod_{i=1}^{n} p(F_i = f_i|C = c) \end{array} \tag{3}$$

The conditional probabilities $p(F_i = f_i|C = c)$ are measured in the learning phase. Because this task is not complicated, we don't explain those calculations in detail. Let us focus on probability $p(C)$. We want to calculate the probability of two random transactions being concluded by the same customer. When the probability of a transaction belonging to customer A appears in the database with probability p_A, we can conclude that two randomly selected transactions belong to the same customer with probability p_A^2. This feature explains why we cannot measure $p(C)$ on the basis of a learning dataset. This dataset is a fraction of a complete database. In consequence, it contains a different number of customers in comparison with the complete dataset. We have designed the following algorithm to measure the average probability $p(C = true)$ that two transactions being concluded by the same customer. Let us assume that the size of the dataset to be clustered equals $sizec$ and the average number of transactions per customer in the learning dataset equals avt. Then the probability $p(C = true)$ is approximated as the multiplication of the average customer number $sizec/avt$ by the probability of the pairs of their transactions being randomly chosen $(avt/sizec)^2$. Summing up the approximated probability $p(C = true)$ equals:

$$p(C = true) = \frac{avt}{sizec} \tag{4}$$

Table 1. Example transactions

Tr_Id	Cl_Id	E-mail	Phone	Name
1	1	clara@domain.com	123456797	Bob Hopkins
2	1	clara@domain.com	123456798	Bob Hopkins
3	1	bob@domain.com	123456798	Bob Hopkins
4	2	adrian@domain.com	123456797	Adrian Jones
5	2	bob@domain.com	987654321	Mary Jones

In contrast to the classifier definition, the distance function operates on clusters not on single transactions. In order to adapt the Naive Bayesian classifier, we have decided to measure: a) the maximum probability *pg* of two clusters A and B contain transactions concluded by the same customer; b) the maximum probability *npg* of two clusters contain transactions concluded by different customers. Then those values are used to evaluate the *similarity* variable.

Let us consider the dataset showed in tab. 2. Columns Tr_Id and Cl_Id represent the transaction key and the cluster key respectively. Cluster 1 and cluster 2 do not have one unique value for attributes $e - mail$ and *phone*. This property prevents a straightforward application of the Naive Bayesian classifier. Having analyzed this dataset, we cannot identify pair (A, B) where A represents a transaction from cluster 1 and B stands for a transaction from cluster 2 which have equal e-mail and phone values. On the other hand, when transactions 4 and 5 are clustered together, it means each value of a given attribute identifies the cluster. As a result, the pair $(bob@domain.com, 123456797)$ identifies cluster 2. This observation leads to the solution, clusters A and B are equal on the level of a given attribute if there is at least one value which exists in both clusters. On the other hand, the dissimilarity distance between clusters equals maximum *npg* between transactions belonging to different clusters.

4 Experiments

At the beginning we have created a simple method which identifies particular customers among selling transactions according to client names and e-mails. This method quality was low because it is not resistant to any variations of textual data. On the other hand, it was sufficient to identify following categories: a) customers with one or a few e-mail addresses; b) e-mail addresses assigned to the group of customers; and c)transactions where e-mail addresses and customer names create complicated relations. We have diagnosed that this method creates too many partitions for categories a) and b). However the most problematic was category c) which creates partitions containing many customers. In consequence, they were useless to any further analysis. Then we have applied the introduced clustering algorithm.

Table 2. Clusterization comparison

Attribute	Test1	Test2
Size	10221	4501
Basic	1526	1732
Cluster	2944	1626

The test was conducted on the computer with 4GB RAM and Intel Core 2 Duo P8600 2.4GHz. Our dataset was a real data set of around 35 000 transactions coming from e-commerce system. We have created two testing data sets. $Test1$ consists of all transactions belonging to the category c). $Test2$ contains 4 501 transactions belonging to categories a) and b). Because partitions belonging to data category a) and b) identify particular customers almost correctly, we have decided to use 4 000 transactions coming from those partitions as a learning dataset without additional processing. Next we have processed datasets $Test1$ and $Test2$. Then the resulting clusters were verified manually by the client if they meet the expectations. Table 2 concludes our results. The $Size$ attribute is the dataset size. Basic and Cluster refers to the number of identified particular customers by means of the basic partitioning algorithm and the agglomerative version respectively.

We can see that our solution achieve both aims. Our cluster algorithm almost doubles the amount of particular customers for category c). It also creates less clusters for datasets a) and b). It is noteworthy that the testing data was extracted automatically which allows us to automatize the whole process. Additionally, we have measured the calculation time of clustarization. The datasets $Test1$ and $Test2$ were processed during 14 minutes and 3 minutes respectively. The fist dataset was processed noticeably longer because it contains more similar transactions. Nevertheless, the time complexity of our algorithm is adequate to the data available in medium sized e-commerce database.

5 Conclusion

There are powerful methods for analyzing the behavior of users in social networks. These methods help drawing conclusions about the attitude of users to products. However, many customer databases are sparsely populated. Even worse, the data in such databases may be ambiguous because the same person uses different proxies for his/her activities. Moreover, the name of a single customer may have been recorded in many slightly different ways. Hence, we need a method that assigns to each customer the data really associated to him/her, using a unique terminology.

The ability to work on real data shows that transaction log from e-commerce has hidden information about customers relationship. Those relations appear

when we analyze e-mails, addresses and phone prefixes because institutions usually have constant elements. On the other hand, people tend to use their own data even when they make transactions for a company.

In the paper, we have proposed the new algorithm which is a composition of agglomerative clustering and the Naive Bayesian classifier. Thanks to it we can uncover unique customers from the concluded transactions. In the future work, we want to optimize this algorithm calculation complexity because it equals $\mathcal{O}(n^2)$ in the worst case. Another interesting aspect of further research is as follows. Having known the relations between customers, we can explain the propagation of customer actions using the terminology of stream processing data. Our experiments in the area of stream database optimizations [15] shows that the partitioning of data processing plan reduces substantially information latency. When customers are assumed as data stream operators then we can use similar strategy to accelerate information propagation in such a graph of customer relationship. In consequence, the communication between customer and e-commerce may gain from better understating of user communication habits with other.

References

1. Asur, S., Huberman, B.A., Szabó, G., Wang, C.: Trends in social media: Persistence and decay. CoRR, abs/1102.1402 (2011)
2. Awadallah, R., Ramanath, M., Weikum, G.: Opinionetit: understanding the opinions-people network for politically controversial topics. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 2481–2484. ACM, New York (2011)
3. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proceedings of the Sixth ACM SIGKDD on Knowledge Discovery and Data Mining, KDD 2000, pp. 407–416. ACM, New York (2000)
4. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA (2002)
5. Gorawski, M.: Architecture of parallel spatial data warehouse: Balancing algorithm and resumption of data extraction. In: Software Engineering: Evolution and Emerging Technologies, pp. 49–59 (2005)
6. Gorawski, M.: Extended cascaded star schema and ECOLAP operations for spatial data warehouse. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 251–259. Springer, Heidelberg (2009)
7. Gorawski, M.: Multiversion spatio-temporal telemetric data warehouse. In: Grundspenkis, J., Kirikova, M., Manolopoulos, Y., Novickis, L. (eds.) ADBIS 2009. LNCS, vol. 5968, pp. 63–70. Springer, Heidelberg (2010)
8. Gorawski, M.: Time complexity of page filling algorithms in materialized aggregate list (mal) and mal/trigg materialization cost. Control and Cybernetics 38(1), 153–172 (2009)
9. Gorawski, M., Bańkowski, S., Gorawski, M.: Selection of structures with grid optimization, in multiagent data warehouse. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 292–299. Springer, Heidelberg (2010)
10. Gorawski, M., Bularz, J.: Protecting private information by data separation in distributed spatial data warehouse. In: ARES, pp. 837–844 (2007)

11. Gorawski, M., Chechelski, R.: Online balancing of aR-tree indexed distributed spatial data warehouse. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Waśniewski, J. (eds.) PPAM 2005. LNCS, vol. 3911, pp. 470–477. Springer, Heidelberg (2006)
12. Gorawski, M., Chrószcz, A., Gorawska, A.: User Identity Unification in e-Commerce. In: Swiątek, J., Grzech, A., Swiątek, P., Tomczak, J.M. (eds.) Advances in Systems Science. AISC, vol. 240, pp. 163–172. Springer, Heidelberg (2014)
13. Gorawski, M., Chrószcz, A.: Query processing using negative and temporal tuples in stream query engines. In: Szmuc, T., Szpyrka, M., Zendulka, J. (eds.) CEE-SET 2009. LNCS, vol. 7054, pp. 70–83. Springer, Heidelberg (2012)
14. Gorawski, M., Chrószcz, A.: Streamapas: Query language and data model. In: CISIS, pp. 75–82 (2009)
15. Gorawski, M., Chrószcz, A.: Optimization of operator partitions in stream data warehouse. In: Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP, DOLAP 2011, pp. 61–66. ACM, New York (2011)
16. Gorawski, M., Dowlaszewicz, K.: Algorithms for efficient top-k spatial preference query execution in a heterogeneous distributed environment. In: ICEIS (1), pp. 43–48 (2009)
17. Gorawski, M., Dyga, A.: Indexing of spatio-temporal telemetric data based on distributed mobile bucket index. In: Parallel and Distributed Computing and Networks, pp. 292–297 (2006)
18. Gorawski, M., Dyga, A.: Multi-dimensional dynamic bucket index based on mobile agent system architecture. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 924–934. Springer, Heidelberg (2006)
19. Gorawski, M., Dyga, A.: Indexing of spatio-temporal telemetric data based on adaptive multi-dimensional bucket index. Fundam. Inform. 90(1-2), 73–86 (2009)
20. Gorawski, M., Faruga, M.: Stah-tree: Hybrid index for spatio temporal aggregation. In: ICEIS (1), pp. 113–118 (2007)
21. Gorawski, M., Gebczyk, W.: Distributed approach of continuous queries with knn join processing in spatial data warehouse. In: ICEIS (1), pp. 131–136 (2007)
22. Gorawski, M., Gorawski, J.I.M.: Algorytm adaptacyjnego balansowania obcienia zapyta w rozproszonych przestrzenno-temporalnych hurtowniach danych. Studia Informatica 32(2A), 75–88 (2011)
23. Gorawski, M., Gorawski, M.: Balanced spatio-temporal data warehouse with r-mvb, stcat and bitmap indexes. In: PARELEC, pp. 43–48 (2006)
24. Gorawski, M., Gorawski, M.: Modified R-MVB tree and BTV algorithm used in a distributed spatio-temporal data warehouse. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 199–208. Springer, Heidelberg (2008)
25. Gorawski, M., Jureczek, P.: Continuous pattern mining using the fCPGrowth algorithm in trajectory data warehouses. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010, Part I. LNCS, vol. 6076, pp. 187–195. Springer, Heidelberg (2010)
26. Gorawski, M., Jureczek, P.: Regions of interest in trajectory data warehouse. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010. LNCS, vol. 5990, pp. 74–81. Springer, Heidelberg (2010)
27. Gorawski, M., Jureczek, P.: Extensions for continuous pattern mining. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 194–203. Springer, Heidelberg (2011)
28. Gorawski, M., Kaminski, M.: On-line balancing of horizontally-range-partitioned data in distributed spatial telemetric data warehouse. In: DEXA Workshops, pp. 273–280 (2006)

29. Gorawski, M., Lis, D.: Architektura cuda w bezopnieniowych hurtowniach danych. Studia Informatica 32(2A), 157–167 (2011)
30. Gorawski, M., Lorek, M., Gorawska, A.: Cuda powered user-defined types and aggregates. In: AINA Workshops, pp. 1423–1428 (2013)
31. Gorawski, M., Lorek, M., Gorawski, M.: Encrypted adaptive storage model – analysis and performance tests. In: Fischer-Hübner, S., Katsikas, S., Quirchmayr, G. (eds.) TrustBus 2012. LNCS, vol. 7449, pp. 118–128. Springer, Heidelberg (2012)
32. Gorawski, M., Malczok, R.: Multi-thread processing of long aggregates lists. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Waśniewski, J. (eds.) PPAM 2005. LNCS, vol. 3911, pp. 59–66. Springer, Heidelberg (2006)
33. Gorawski, M., Malczok, R.: AEC algorithm: A heuristic approach to calculating density-based clustering *eps* parameter. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 90–99. Springer, Heidelberg (2006)
34. Gorawski, M., Malczok, R.: Calculation of density-based clustering parameters supported with distributed processing. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 417–426. Springer, Heidelberg (2006)
35. Gorawski, M., Malczok, R.: Materialized ar-tree in distributed spatial data warehouse. Intell. Data Anal. 10(4), 361–377 (2006)
36. Gorawski, M., Malczok, R.: Towards automatic *eps* calculation in density-based clustering. In: Manolopoulos, Y., Pokorný, J., Sellis, T.K. (eds.) ADBIS 2006. LNCS, vol. 4152, pp. 313–328. Springer, Heidelberg (2006)
37. Gorawski, M., Malczok, R.: Towards stream data parallel processing in spatial aggregating index. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 209–218. Springer, Heidelberg (2008)
38. Gorawski, M., Malczok, R.: Answering range-aggregate queries over objects generating data streams. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5982, pp. 436–439. Springer, Heidelberg (2010)
39. Gorawski, M., Malczok, R.: Cam2s: An integrated indexing structure for spatial objects generating data streams. In: CISIS (2010)
40. Gorawski, M., Malczok, R.: Indexing spatial objects in stream data warehouse. In: Nguyen, N.T., Katarzyniak, R., Chen, S.-M. (eds.) Advances in Intelligent Information and Database Systems. SCI, vol. 283, pp. 53–65. Springer, Heidelberg (2010)
41. Gorawski, M., Marks, P.: Resumption of data extraction process in parallel data warehouses. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Waśniewski, J. (eds.) PPAM 2005. LNCS, vol. 3911, pp. 478–485. Springer, Heidelberg (2006)
42. Gorawski, M., Marks, P.: Checkpoint-based resumption in data warehouses. In: Sacha, K. (ed.) Software Engineering Techniques: Design for Quality. IFIP, vol. 227, pp. 313–323. Springer, Boston (2006)
43. Gorawski, M., Marks, P.: Fault-tolerant distributed stream processing system. In: DEXA Workshops, pp. 395–399 (2006)
44. Gorawski, M., Marks, P.: Distributed stream processing analysis in high availability context. In: ARES, pp. 61–68 (2007)
45. Gorawski, M., Marks, P.: Towards reliability and fault-tolerance of distributed stream processing system. In: DepCoS-RELCOMEX, pp. 246–253 (2007)
46. Gorawski, M., Marks, P.: Towards automated analysis of connections network in distributed stream processing system. In: Haritsa, J.R., Kotagiri, R., Pudi, V. (eds.) DASFAA 2008. LNCS, vol. 4947, pp. 670–677. Springer, Heidelberg (2008)
47. Gorawski, M., Marks, P., Gorawski, M.: Collecting data streams from a distributed radio-based measurement system. In: Haritsa, J.R., Kotagiri, R., Pudi, V. (eds.) DASFAA 2008. LNCS, vol. 4947, pp. 702–705. Springer, Heidelberg (2008)

48. Gorawski, M., Morzy, T., Wrembel, R., Zgrzywa, A.: Advanced data proceedings and analysis techniques. Control and Cybernetics 40, 581–583 (2012)
49. Gorawski, M., Panfil, S.: A system of privacy preserving distributed spatial data warehouse using relation decomposition. In: ARES, pp. 522–527 (2009)
50. Gorawski, M., Pasterak, K.: Agkpstream a operatory strumieniowe. Studia Informatica 33(2A), 181–195 (2012)
51. Gorawski, M., Pasterak, K.: Schedulery strumieniowe w agkpstream. Studia Informatica 33(2A), 197–210 (2012)
52. Gorawski, M., Pluciennik, E.: Distributed data mining by means of SQL enhancement. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 34–35. Springer, Heidelberg (2008)
53. Gorawski, M., Płuciennik-Psota, E.: Distributed data mining methodology with classification model example. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 107–117. Springer, Heidelberg (2009)
54. Gorawski, M., Siedlecki, Z.: Implementation, optimization and performance tests of privacy preserving mechanisms in homogeneous collaborative association rules mining. In: Meersman, R., et al. (eds.) OTM 2011, Part I. LNCS, vol. 7044, pp. 347–366. Springer, Heidelberg (2011)
55. Gorawski, M., Siedlecki, Z.: Optimization of privacy preserving mechanisms in homogeneous collaborative association rules mining. In: ARES, pp. 347–352 (2011)
56. Gorawski, M., Stachurski, K.: On efficiency and data privacy level of association rules mining algorithms within parallel spatial data warehouse. In: ARES, pp. 936–943 (2006)
57. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD 1998, pp. 73–84. ACM, New York (1998)
58. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)
59. Heller, K.A., Ghahramani, Z.: Bayesian hierarchical clustering
60. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3), 264–323 (1999)
61. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream
62. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the Sixth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2000, pp. 169–178. ACM, New York (2000)
63. Olson, C.F.: Parallel algorithms for hierarchical clustering. Parallel Computing 21, 1313–1325 (1993)
64. Walter, B., Bala, K., Kulkarni, M., Pingali, K.: Fast agglomerative clustering for rendering. In: IEEE Symposium on Interactive Ray Tracing (RT), pp. 81–86 (August 2008)

Network Management Based on Domain Partition for Mobile Agents

Yonghui Liu and Weidong Min

School of Computer Science & Software Engineering,
Tianjin Polytechnic University, Tianjin, 300387, China
minweidong@tjpu.edu.cn

Abstract. Combining with SNMP and SOAP web service technology, this paper proposes a mobile agent based hierarchical network management framework to alleviate the scalability limitation of centralized network management model. In the proposed solution, multiple mobile agents are dispatched by management station to perform the management tasks simultaneously. The paper discusses how to partition the managed devices into proper number of domains while each of the domains is load balanced. The upper bound of the number of domains is given. A Max-Min adjustment algorithm is proposed to balance the load of every domain. Mathematical analysis and experimental results show that the proposed solution has better performance in both network traffic and response time than the centralized network management solution.

Keywords: Network management, Mobile agent, Domain partition, SNMP.

1 Introduction

Network resource management is critical to ensure the security and optimal performance of distributed networks and information systems. Existing network management models are mostly based on SNMP, in a centralized client / server architecture and widely deployed in IP-based network [1]. However, the existing methods are not suitable in increasingly large scale network management, where the SNMP involves massive transfers of management data between a management server which is called as a management station, and the network elements to be managed which are called as managed devices. The large amount of network throughput and intensive processing time at the management station has become a management bottleneck.

Mobile Agent (MA) with the distributed intelligent management approach has been proposed to overcome the above problems in the past few years [2-5]. Because of its independence and mobility, mobile agent can reduce the network traffic, by migrating to the location of the managed devices to manage them locally and therefore relieve the burden of the management station. MA size increases as mobile agent moves from one host to another to perform management activities. The bloated MA increases the migration costs and has scalability problem.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 153–160, 2013.

In this paper, a mobile agent based hierarchical network management model is proposed to alleviate the above mentioned scalability limitation of mobile agent. In this model, the managed devices are partitioned into several domains. Mobile agents are dispatched by the management station to manage the devices. Instead of returning themselves to the management station, mobile agents pass back the management data to the management station using SOAP, which can reduce the migration costs of mobile agents. SOAP is a secure communication protocol and widely used both in industry and academic fields [6].

The remainder of this paper structures as follows. In Section 2, the related work is discussed. Section 3 describes the architecture of the proposed network management framework. Theoretical analysis and performance compare of the proposed model versus centralized network management model is discussed in Section 4. Section 5 discusses the management domain partition optimization and algorithm. Experimental results are given in Section 6, followed by conclusions in Section 7.

2 Related Works

Mobile agent has been proposed for the management of networks and distributed systems. Al-Kasassbeh et al. [3] discussed various mobile agent based network management models, analyzed the key advantages of mobile agent in distributed network management and showed that mobile agent could alleviate bandwidth overloading problem. In paper of Holt et al. [4], performance analysis of mobile agent showed that optimal performance could be obtained when a small number of mobile agents sharing the management tasks. Similar study was showed by Verma et al. [5], where they partitioned the travel itinerary of MA so that multiple MAs can work in parallel.

For large network domain, by partitioning the managed devices into several small domains, management tasks can be performed in parallel. Thus, the network management efficiency can be improved. Huang et al. [7] investigated a clustering strategy that a large domain was divided into a number of clusters. Multiple MAs performed the management tasks simultaneously. The results analysis showed that this approach can reduce the processing time of management station. In order to overcome the inefficiency of the 'flat' network structure in agent based management framework, Gavalas et al. [8] segmented the network and deployed multiply middle level managers for monitoring managed devices.

However, neither of the above partition strategies gave the explicit number of the clusters or segments. What's more, in those papers, they partitioned N managed devices into K domains, each of which contained N/K devices. However, this method did not take the number of tasks into account, where each of the managed devices may contains different number of tasks. The domains may not be load balanced when tasks in the devices are not equally distributed. Differently from these approaches, in this paper a domain partition approach based on the number of tasks is proposed, by which every domains can be better load balanced.

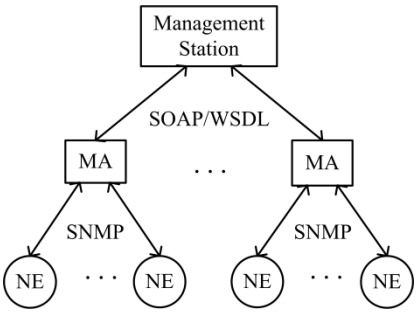


Fig. 1. The architecture of mobile agent based hierarchical network management model

3 The Proposed Mobile Agent Based Hierarchical Network Management Framework

The mobile agent based hierarchical network management framework proposed in this paper, consists of three major components, as illustrated in Fig. 1.

(1) Management Station (MS): this is a management server. The MS plays a role of managing the mobile agents, dispatching them to the remote hosts, suspending or abandoning them when it is necessary. MS receives the management data from mobile agents and records them to the database.

(2) Domain Manager (DM): this is a host where the mobile agent stays on. Usually, the host that has high processing capacity is chosen as the domain manager.

(3) Mobile Agent (MA): it is an executable program staying on the DM. The MA is dispatched and controlled by the management station. MA undertakes the full responsibility to manage the devices within the domain. By SNMP, the MA gets the monitoring information from the managed device and returns them to the MS using SOAP.

In the proposed framework, devices that need monitoring are partitioned into proper number of domains. DMs are selected with each of the domains. Mobile agents are dispatched by MS to each of the DM. Mobile agents manage devices using SNMP within the domain and transfer the results to the MS using SOAP web service. Through domain partition, multiple mobile agents can conduct the management tasks in parallel, which can improve the efficiency of network management.

4 Performance Analysis of the Proposed Network Management Model

4.1 Network Traffic Analysis

We evaluate the network traffic generated by the MS when performing the management activities. In our proposed model, Mobile agents retrieve the MIB objects from the managed devices and return results to the MS.

Suppose N managed devices and Q MIB objects are to be retrieved in the network. The whole network is partitioned into K domains, each of the domains contains Q_j MIB objects ($Q = \sum_{j=1}^{K} Q_j$, $1 \leq j \leq K$). The average size of one MIB object can be expressed as S_{data}. Mobile agent queries those data and passes them back to the MA using SOAP. We use S_{sop} as the average size of protocol related packets. The traffic generated for transferring the Q_j MIB objects to the MS, can be expressed as $T_{MA}(j) = Q_j * S_{data} + S_{sop}$. So, the bandwidth consumed around the MS in the proposed model in once polling period is as follows:

$$T_{MA} = \sum_{j=1}^{K} \left(Q_j * S_{data} + S_{sop} \right) \tag{1}$$

In centralized management model, the MS sends out a Get Request data S_{req} to the managed device. The device that receives the query will send Get Response data S_{res} to the MS. The total traffic that the MS generates for querying the management data from the managed devices can be calculated as follows:

$$T_{SNMP} = Q * (S_{req} + S_{res}) \tag{2}$$

where S_{req} S_{res} represents the packet average size (bytes) of Get Request and Get Response, respectively.

Compared with Equation (1) and (2), when K is smaller than a certain value, the traffic generated by the MS in the proposed model is less than the traffic generated in the centralized approach because of $S_{data} \ll (S_{req} + S_{res})$.

4.2 Network Response Time Analysis

The response time is the time taken by MS to conduct management tasks. In the proposed model, tasks in different domains are managed simultaneously. The time taken by MS to complete all the tasks in the network is the maximum of time spent to manage tasks in each domain. The time taken by MS to conduct tasks in domain j includes the time for querying MIB objects from the managed devices and the time for results transferring back to MS by mobile agent.

The time $L_{soap}(j)$ taken by mobile agent to return the query results to the management in domain j is:

$$L_{soap}(j) = L_{tran}(j) + L_{prop} + L_{cpu}(j) \tag{3}$$

where $L_{tran}(j)$ represents the average data transmission delays when transferring the data in the WAN, L_{prop} is the network propagation delay and $L_{cpu}(j)$ is the CPU processing delay. $1 \leq j \leq K$. So, the total time taken by MS is:

$$L_{MS} = \max\{ Q_j * L_{query} + L_{soap}(j) \} \tag{4}$$

where L_{query} represents the average execution time taken to retrieve one MIB object.

In centralized network management model, the total time taken by the MS to query Q MIB objects is given by the following expression:

$$L_{SNMP} = Q * (L_{query} + L_{latcy}) \tag{5}$$

where, L_{latcy} represents the propagation delay in WAN.

Compared with Equation (4) and (5), the time taken by MS to perform the management tasks in the proposed model is less than the time taken in the centralized management model.

5 Management Domain Partition

5.1 The Selection Strategy for the Number of Domains

A priority issue for management domain partition is to decide the number of required domains. Based on Equation (4), when the number of tasks is fixed the more of the number of domains, the smaller of the number of tasks in each domain and the task completion time reduces accordingly. However, more domains may generate more maintenance costs.

Based on Equation (1) and (2), the performance of the proposed model is inferior to the performance of the centralized management model, when the number of mobile agent reaches to a threshold value. We get the upper bound of the value K, $K \leq Q * \frac{(S_{req}+S_{res}-S_{data})}{S_{sop}}$, which is used in the management domain partition algorithm as a constraint condition.

5.2 Management Domain Partition Algorithm

Managed devices in the network containing different number of tasks should be partitioned into the domain evenly so that all the domains can be load balanced. Suppose n devices are to be partitioned into k domains where D_j is a set that contains the managed devices partitioned into domain j and S_j is the number of managed devices partitioned into domain j, $1 \leq j \leq k$. Mobile agent residing on the domain manager manages the managed devices. Parameter $e_{i,j}$ represents the time that mobile agent takes to manage device i in domain j, $1 \leq i \leq n$. The execution time matrix $E_{n\times k}$ is a matrix consisting of $e_{i,j}$. The time taken by mobile agent to manage all the S_j devices in domain j can be calculated as $T_j = \sum_{r=1}^{S_j} e_{i,r}$. The object function is

$$f_{obj} = min\left(max_{1 \leq j \leq n}(T_j)\right) \tag{6}$$

The constraint function can be expressed as:

$$f_{bal} = \sqrt{\frac{1}{k}\sum_{j=1}^{k}\left(T_j - \bar{T}\right)^2} \tag{7}$$

where $\bar{T} = \frac{1}{k}\sum_{j=1}^{k} T_j$.

The Max-Min adjustment algorithm is proposed to solve this problem, which includes two phases. First the network elements are grouped into k domains based on the minimize execution time $e_{i,j}$ in time matrix $E_{n\times k}$. Then the domain is adjusted to

be load balanced through changing the elements from large load domains to small load domains. The following procedure describes the Max-Min adjustment algorithm:

Step 1. In every row i of time matrix $E_{n\times k}$ find out the minimum time $e_{i,j}$, then add i into set D_j, where D_j is the set of devices which are partitioned into domain j.

Step 2. Do the following **Step 3**, **Step 4**, **Step 5** and **Step 6** until the constraint function f_{bal} is less than the given value.

Step 3. For every set D_j, calculate the execution time $T_j = \sum_{r=1}^{S_j} e_{i,j}$, then find out the maximum T_u and minimum T_v, which indicate that D_u and D_v have the maximum and minimum load among all the domains. We adjust them by the following steps.

Step 4. For every device r in domain D_u calculate the costs of changing device r from domain D_u to domain D_v by using $c_r = e_{r,v} - e_{r,u}$, where $e_{r,v}$ and $e_{r,u}$ represent the time taken by mobile agent to manage device r in domain D_u and D_v, respectively.

Step 5. Find out the minimum c_r, then delete the device r from domain D_u and add it to domain D_v.

Step 6. Go to **Step 2** to continue.

6 Experiments and Results

The proposed network management solution based on domain partition for mobile agents has been implemented. Mobile agent with the ability to retrieve data from the designated PCs was programmed using C++. The communication between mobile agents and the MS using SOAP was developed in Java. The Ethereal was used as the tool to capture and filter the packets that we needed. The system time of PCs was used to count the task completion time.

In order to evaluate the performance of the proposed management model and the centralized management model, several experiments were done. In those experiments, the MS was deployed in a LAN, while the managed devices were deployed in a remote LAN. The bandwidth of the LAN was 10 Mb/s. Both LANs were connected via a wide area network using routers. 200, 400, 600, 800 and 1000 tasks on 100 PCs were used to represent the size of the network domain.

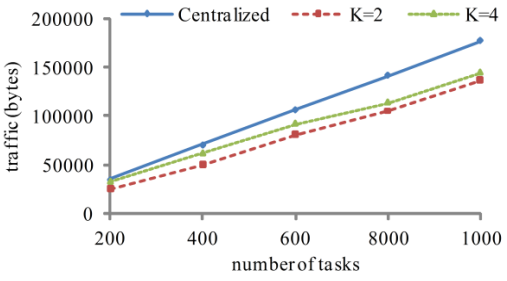


Fig. 2. The traffic compare between the proposed model and the centralized model

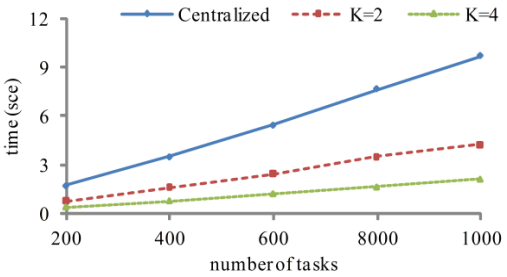


Fig. 3. The time compare between the centralized model and the proposed model

The traffic generated by the MS and the response time taken by MS in the proposed model were measured, where the domain is partitioned into 2 domains, 4 domains and in a centralized model, respectively. The experimental results are showed in Fig.2 and Fig.3. The results in Fig.2 show that the proposed model can reduce the network traffic compared with the centralized management model. From Fig.3 we can see that the response time decreases in the proposed model compared with the centralized management model.

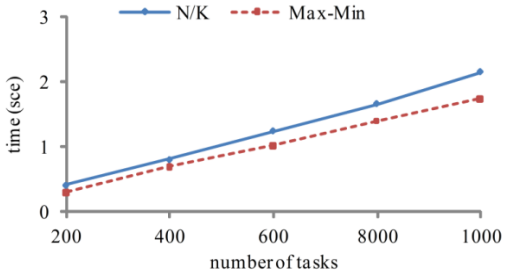


Fig. 4. The tasks completion time compare between the Max-Min adjustment algorithm and the N/K approach

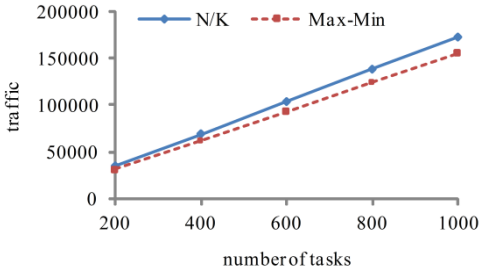


Fig. 5. The traffic compare between the Max-Min adjustment algorithm and the N/K approach

Our Max-Min adjustment algorithm was also compared with an even domain partition N/K method of paper [8], using 20, 40, 60, 80 and 100 PCs. The number of MIB objects on these devices ranged from 1 to 20. We randomly generated these numbers and got the total number of tasks 200, 400, 600, 800 and 1000 as the size of the domain. Fig.4 and Fig.5 indicate that the Max-Min adjustment algorithm can shorten the tasks completion time and generate less traffic compared with the N/K approach.

7 Conclusions

In this paper, a network management approach based on domain partition for mobile agents has been proposed, using the Max-Min adjustment algorithm to partition the management domain. A quantitative evaluation, in terms of the network traffic and the response time in the proposed model has been discussed. Experimental results showed that the proposed network management model has better performance in both network traffic and response time than the centralized network management solution.

Acknowledgements. This research is supported by the Natural Science Foundation of Tianjin, China under grant No. 13JCYBJC15500.

References

1. Zhang, W.W., Chen, S.Y.: Design and Implementation of SNMP-Based Web Network Management System. J. Advanced Materials Research 341, 705–709 (2012)
2. Min, W., Chen, K., Ke, Y.: A Matrix Grammar Approach for Automatic Distributed Network Resource Management. J. Frontiers of Computer Science (accepted, in press, 2013)
3. Al-Kasassbeh, M., Adda, M.: Analysis of mobile agents in network fault management. J. Journal of Network and Computer Applications. 31(4), 699–711 (2008)
4. Holt, A., Huang, C.Y., Monk, J.: Performance analysis of mobile agents. J. IET Communications 1(3), 532–538 (2007)
5. Verma, V.K., Joshi, R.C., Xie, B., Agrawal, D.P.: Combating the bloated state problem in mobile agents based network monitoring applications. J. Computer Networks 52(17), 3218–3228 (2008)
6. Chourmouziadis, A., Charalambides, M., Pavlou, G.: On the Performance and Scalability of Web Services for Monitoring MPLS-based Networks. J. Journal of Network and Systems Management 17(1-2), 105–136 (2009)
7. Huang, C.Y., Cheng, K., Holt, A.: An integrated manufacturing network management framework by using mobile agent. J. The International Journal of Advanced Manufacturing Technology 32(7-8), 822–833 (2007)
8. Gavalas, D., Politi, C.T.: Low-cost itineraries for multi-hop agents designed for scalable monitoring of multiple subnets. J. Computer Networks 50(16), 2937–2952 (2006)

Multi-objective Quantum Cultural Algorithm and Its Application in the Wireless Sensor Networks' Energy-Efficient Coverage Optimization

Yi-nan Guo, Meirong Chen, and Chun Wang

China University of Mining and Technology, Xuzhou, Jiangsu, China

Abstract. Whether wireless sensor network covers the target field availably is measured by the network cover rate and the node redundancy rate. To solve this multi-objective optimization problem, multi-objective quantum cultural algorithm is proposed. It effectively utilizes the implicit knowledge extracted from the non-domination individuals to promote more efficient search. Two highlights include: 1. The rectangle's height of each allele is calculated in accordant with the non-dominated sort among individuals. 2. The update operation of quantum individuals and the mutation operator are directed by the implicit knowledge. Taken a typical wireless sensor network with 25 sensor nodes as an example, simulation results indicate that the layout of wireless sensor network obtained by the proposed algorithm have larger network cover rate and lower node redundancy rate.

Keywords: Wireless sensor network, cultural algorithm, knowledge, real-coded quantum evolutionary algorithm, multi-objective optimization.

1 Introduction

Wireless sensor network(WSN) containing large numbers of sensor nodes monitors the targets. Its energy-efficiency coverage optimization is actually how to rationally arrange the sensor nodes. Thus it is converted to a multi-objective optimization problem(MOP). Now, many natural-inspired optimization algorithms are introduced to solve the problems, such as genetic algorithm[1], ant colony optimization[2], particle swarm algorithm[3] and quantum algorithm[4]. However, the optimization objective normally is the network coverage rate only or the weighted objective combing the network coverage rate and node redundancy rate. The conflict between objectives was not considered. So we regard maximum network coverage rate and minimum node redundancy rate as two independent objectives.

Until now, many multi-objective optimization algorithms had been presented, such as strength pareto evolutionary algorithm II(SPEAII)[5], no-dominated sorting genetic algorithm II(NSGAII)[6] and multi-objective quantum evolutionary algorithms(MOQA). Meshoul[7] described the chromosome by quantum-encoded probabilistic representation. The principles of quantum computing including superposition and interference[8] are employed to improve the quality of the non-dominated solution set. After decomposing MOP into many scalar sub-problems, each sub-problem

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 161–167, 2013.

simultaneously evolved based on the population composed of *q*-bit individuals[9]. A triploid chromosome and the chaos encoding probability amplitude are constructed[10]. However, they did not fully utilize the implicit information in the evolution, which limits the algorithm's performances. Based on the dual structure in cultural algorithm[11], the author gave real-coded quantum cultural algorithm[12]. But it only fits for the scalar optimization problem. So a novel multi-objective real-coded quantum cultural algorithm(MOQCA) is proposed and applied to optimize WSNs' energy-efficiency coverage.

2 The WSN' Energy-Efficient Coverage Optimization Model

The key issues that directly influence WSN' energy-efficiency coverage model lie on two aspects: (i)the sensor node's sensing model. (ii)how to evaluate the performances of WSN' energy-efficiency. Suppose there are *N* sensor nodes.

The sensor nodes' sensing model describes its monitoring ability and sensing range. In this paper, the probability sensing model is adopted, in which the probability for monitoring the target o_k by the sensor node $s_i=\{x_i, y_i\}$, expressed by $P_{\text{cov}}(o_k, s_i)$, is exponentially decreased as the distance between them is increasing[13]. As all nodes contribute to monitor the target, the ability to monitor o_k decides by the joint probability of all nodes, defined as $P_{\text{cov}}(o_k)=1-\prod_{i\in N}(1-P_{\text{cov}}(o_k, s_i))$.

The QoS of WSN is commonly measured by the network coverage rate and the node redundancy rate. The former reflects the coverage degree of monitored area by all sensor nodes. The latter measures the uniformity degree on sensor nodes' distribution. Without loss of generality, the monitored area is evenly partitioned into discrete grid along X-Y coordination[13]. Both objectives are obtained based on the joint monitored probability of all sensor nodes:

$$f_1 = NCR = \frac{\sum_{k=1}^{a\times b} P_{\text{cov}}(o_k)}{a\times b} \tag{1}$$

$$f_2 = NRR = \frac{N\pi r^2 - \sum_{k=1}^{a\times b} P_{\text{cov}}(o_k)}{N\pi r^2} \tag{2}$$

The essential of above optimization problem is to obtain an optimal sensor nodes' arrangement, which has maximum NCR and minimum NRR. Two objectives are incompatible each other. In order to simply the computation, we covert all objectives to maximum problem.

3 The WSN's Energy-Efficiency Coverage Optimization Method Based on MOQCA

3.1 MQEA in Population Space

There are two kinds of individuals in MQEA: evolutionary individuals and quantum individuals. Each evolutionary individual denotes the sensor nodes' locations as

$p^j(t)=\{s_1^j,s_2^j,\cdots,s_N^j\}=\{(x_1^j,y_1^j),(x_2^j,y_2^j)\cdots,(x_N^j,y_N^j)\},i=1,2,\cdots n$. n is population size. Each gene of $p^j(t)$ is described by a rectangle in a quantum individual $q^j(t)$, which is uniquely determined by its center and width[14] expressed by $(x_{cj}^i, x_{wj}^i, y_{cj}^i, y_{wj}^i)$. Because non-dominated rank can evaluate individuals instead of the fitness values, we present a novel method to calculate the rectangle's height of $q^j(t)$ in MOP based on non-dominated sorting method. Let $zc^i(t)$ be the number of dominating individuals.

$$x_{hj}^i(t)=\frac{zc^i(t)+1}{\sum_{k=1}^{n}(zc^k(t)+1)} \tag{3}$$

3.2 The Extraction and Utilization of Knowledge in Belief Space

In this paper, normative knowledge and situational knowledge are used. Normative knowledge expressed by $K_1=\langle(l_{x1}(t),u_{x1}(t)),(l_{y1}(t),u_{y1}(t)),(l_{x2}(t),u_{x2}(t)),(l_{y2}(t),u_{y2}(t)),\cdots,(l_{xN}(t),u_{xN}(t)),(l_{yN}(t),u_{yN}(t))\rangle$ memorizes the feasible search space of optimization problems. It provides a set of variable ranges and directs the evolutionary individuals to the recorded search space so as to avoid the unnecessary search. u_{xi} and l_{xi} respectively are the upper and lower bounds of *i*th sensor node in *X* coordinate. Normative knowledge directs the mutation operation of evolutionary individuals so as to avoid the blind selection. In multi-objective optimization problem, two copies are created via mutation operation on two parent individuals each time so as to ensure the offspring individuals' diversity. We choose $e_j^k(t),k=\arg\min_l\left\|e^l(t)-p^i(t)\right\|$ as the reference individuals for the mutation operator. Then *i*th non-dominated individual $x_j^i(t)$ is mutated.

$$\overline{x_j^i}(t)=\begin{cases}x_j^i(t)+\lambda(e_j^k(t)-x_j^i(t)) & x_j^i(t)\in[l_{xj}(t),u_{xj}(t)]\\ l_{xj}(t)+\lambda(u_{xj}(t)-l_{xj}(t)) & x_j^i(t)\notin[l_{xj}(t),u_{xj}(t)]\end{cases} \tag{4}$$

Situational knowledge $K_2(t)=<e^1,e^2,\cdots e^i,\cdots e^K>$ records the evolutionary individuals lying in Pareto front during the evolution[14]. *K* is the capability of situational knowledge. The optimal individuals recorded by situational knowledge are sorted in terms of their non-dominated rank and the crowding distance. The update operation of quantum individuals' width and center are influenced by situational knowledge so as to lead the population close to the better area. Suppose θ is the constriction factor of step size. $\delta^i(t)$ denotes the evolution degree and is defined in Section 4. If an evolutionary individual's performances become worse, the rectangle's width is enlarged so as to expand the feasible search space.

$$x_{cj}^i(t+1)=e_{2j-1}^l(t) \tag{5}$$

$$x_{wj}^i(t+1)=x_{wj}^i(t)+\theta\lambda(u_{xj}(t)-l_{xj}(t))\delta^i(t) \tag{6}$$

4 Analysis of the Simulation Results

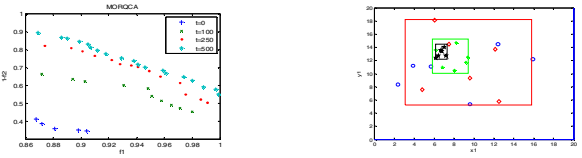
Suppose the size of the monitored area is 20×20. Each sensor node is described by the probability sensing model with following parameters: $r_e=1.5$, $\alpha_1=\beta_1=1$, $\alpha_2=0$, $\beta_2=0.5$, $r=3$. The main parameters in MOQCA and other compared algorithms are:

$N=25$, p_m=0.1, $n=40$, $n_s=20$, T=1000, $n_q=40$, $c=30$, τ=5. In order to quantitatively compare the algorithms' performances, Three metrics[15] including spacing(SP), generation distance(GD) and purity(P) are adopted. Based on GD-metric and SP-metric, the evolution degree is defined as follows. $\delta^i(t)$ is less if the convergence and uniformity of the solutions in last generation are both improved.

$$\delta^i(t)=\begin{cases}-1 & (SP(p^i(t))<SP(p^i(t-1)))\wedge(GD(p^i(t))<GD(p^i(t-1)))\\ 1 & (SP(p^i(t))>SP(p^i(t-1)))\wedge(GD(p^i(t))>GD(p^i(t-1)))\\ 0 & otherwise\end{cases} \tag{7}$$

4.1 The Analysis of the MOQCA's Evolution Process

From the MOQCA's evolution process in any one run time shown in Fig.1, we can know that the network coverage rate is larger and the node redundancy rate is lower along the evolution. Fig.1(a) shows the distribution of non-dominated solutions in the evolution, which indicates the optimal solutions are uniformly distributed in Pareto front gradually. Corresponding knowledge evolution process is recorded in Fig.1(b). Situational knowledge in t=0,100,250,500 are respectively signed by ○, ◇, * and ☆. Corresponding normative knowledge is graphed by the rectangles with the same color. Obviously, situational knowledge is gradually close to the pareto front until the individuals are no longer updated. The recorded decision space varies and gradually covers the better area.



(a) the parto optimal solutions (b) the knowledge evolution

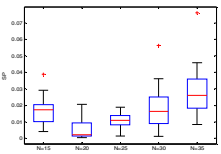
Fig. 1. The evolution process of MOQCA

4.2 Comparison of the Algorithm's Performances with Different *N*

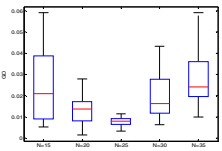
Under N=15, 20, 25, 30, 35, the statistical data are shown in Table.1 and Fig.2. μ and σ are the mean and the standard deviation. Obviously, when N=20, the SP-metric is lower. As the number of sensor nodes is more or less, the algorithm's performance becomes worse. Moreover, GD-metric is less as N=25, which means the pareto font has better convergence and distribution. So we choose N=25 in following experiments. From Fig.2(c), if the network consists of less nodes, all nodes are uniformly arranged, whereas the network coverage rate is not good enough. More sensor nodes can ensure the expected network coverage rate whereas they repeatedly cover the detected area. This increases the network's cost.

Table 1. Comparison of the performances with different number of sensor nodes

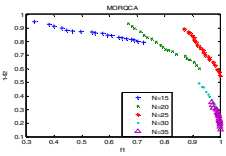
N		15	20	25	30	35
SP	μ	1.74E-2	6.24E-3	1.09E-2	1.85E-2	2.96E-2
	σ	9.74E-3	6.02E-3	4.15E-3	1.42E-2	1.67E-2
GD	μ	2.49E-2	1.36E-2	7. 86E-3	2.01E-2	2.94E-2
	σ	1.87E-2	7.63E-3	2.32E-3	1.11E-2	1.43E-2



(a) SP-metric



(b) GD-metric



c) the parto optimal solutions

Fig. 2. Comparison of the metrics with different number of sensor nodes

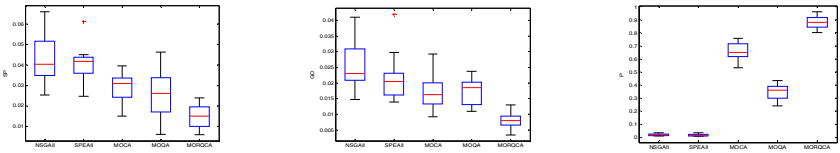
4.3 Comparison of Different Algorithms

Under the same experimental conditions, simulation results derived from applying NSGAII, SPEAII, multi-objective cultural algorithm(MOCA), MOQA, and MOQCA are listed in Table.2 and presented in Fig.3 by box plot.

Table 2. Comparison of the metrics with different algorithms

algorithm		NSGAII	SPEAII	MOCA	MOQA	**MOQCA**
SP	μ	4.32E-2	4.03E-2	2.50E-2	2.61E-2	**1.47E-2**
	σ	1.24E-2	8.15E-3	5.29E-3	1.12E-2	**5.35E-3**
GD	μ	2.51E-2	2.13E-2	1.71E-2	1.73E-2	**7.98E-3**
	σ	7.37E-3	7.13E-3	5.82E-3	4.27E-3	**2.57E-3**
P	μ	2.16E-2	1.72E-2	6.58E-1	3.46E-1	**8.82E-1**
	σ	7.21E-3	7.67E-3	6.61E-2	6.05E-2	**4.86E-2**

The data in Table.2 show that SP-metric and GD-metric of MOQCA are better than other algorithms. The reason for that is in MOQCA, the extracted knowledge is fully used to ensure the evolution toward the potential domination area and make the optimal non-dominated solutions quickly close to the better pareto-optimal front. Besides, the knowledge-inducing mutation operator ensures the better individuals reserved in next generation and avoid the sightless selection. The superposition operation among individuals also increases the optimal individuals' observed probability.



(a) SP-metric (b) GD-metric (c) P-metric

Fig. 3. Comparison of the metrics with different algorithms

5 Conclusions

We choose maximum network cover rate and minimum node redundancy rate as the indices to measure WSN's energy-efficiency coverage and convert it into a two-objective optimization problem. Thereby a novel multi-objective quantum cultural algorithm is proposed to solve this problem. In population space, the rectangle's height of each allele is calculated based on the number of dominated individuals. In belief space, the extracted implicit knowledge records the information about the pare-to-optimal solutions in decision space and utilized to direct the update operation for quantum individuals and the mutation operator. Experimental results show that the proposed algorithm obtained the better and more uniform pareto-optimal solutions by comparing with other multi-objective optimization algorithms. That means the layout of wireless sensor network obtained by the proposed algorithm have larger network cover rate and less node redundancy rate.

Acknowledgments. This work was supported by Jiangsu Natural Science Foundation under Grant BK2010183, Jiangsu Overseas Research & Training Program for University Prominent Young & Middle-aged Teachers and Presidents. Thank you for the help from CERCIA, University of Birmingham.

References

1. Jia, J., Chen, J., Chang, G.-R., et al.: Optimal coverage scheme based on genetic algorithm in wireless sensor networks. Control and Decision 22(11), 1289–1292 (2007)
2. Lee, J.-W., Choi, B.-S., Lee, J.-J.: Energy-efficient coverage of wireless sensor networks using ant colony optimization with three types of pheromones. IEEE Transactions on Industrial Informatics 7(3), 419–427 (2011)
3. Ab Aziz, N.A.B., Mohemmed, A.W., Alias, Y.: A wireless sensor network cverage optimization algorithm based on particle swarm optimization and voronoi diagram. In: IEEE International Conference on Networking, Sensing and Control, pp. 602–607 (2009)
4. Fu, H., Han, S.: Optimal sensor node distribution based on the new quantum genetic algorithm. Chinese Journal of Sensors and Actuators 21(7), 1259–1263 (2008)

5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2:improving the strength Pareto evolutionary algorithm,Technical Report 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zurich, Switzerland (2001)
6. Deb, K.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation 6(2), 182–197 (2002)
7. Meshoul, S., Mahdi, K., Batouche, M.: A quantum inspired evolutionary framework for multi-objective optimization. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 190–201. Springer, Heidelberg (2005)
8. Kim, Y., Kim, J.-H., Han, K.-H.: Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems. In: 2006 IEEE Congress on Evolutionary Computation, pp. 9151–9156 (2006)
9. Wei, X., Fujimura, S.: Multi-objective quantum evolutionary algorithm for discrete multi-objective combinational problem. In: Proceeding of International Conference on Technologies and Applications of Artificial Intelligence, pp. 39–46 (2010)
10. Yang, X.-W., Shi, Y.: A real-coded quantum clone multi-objective evolutionary algorithm. In: Proceeding of International Conference on Consumer Electronic, Communications and Networks, pp. 4683–4687 (2011)
11. Reynolds, R.G.: An introduction to cultural algorithms. In: Proceeding of the Third Annual Conference on Evolutionary Programming, pp. 131–139 (1994)
12. Guo, Y.-N., Liu, D., Cheng, J., et al.: A novel real-coded quantum-inspired cultural algorithm. Journal of Central South University 42, 130–136 (2011)
13. Li, S.J., Xu, C.F., Pan, Y.H.: Sensor deployment optimization for detecting maneuvering targets. In: Proceedings of International Conference on Information Fusion, pp. 1629–1635 (2005)
14. Cruz, A.V.A., Vellasco, M.B.R., Pacheco, M.A.C.: Quantum-inspired evolutionary algorithm for numerical optimization. In: Proceeding of IEEE Congress on Evolutionary Computation, pp. 19–37 (2006)
15. Bandyopadhyay, S., Pal, S.K., Aruna, B.: Multiobjective GAs, quantitative indices and pattern classification. IEEE Transactions on Systems, Man, and Cybernetics-Part B:cybernetics 5(34), 2088–2099 (2004)

Image Super Resolution via Visual Prior Based Digital Image Characteristics*

Yusheng Jia[1], Wanqi Yang[1], Yang Gao[1], Hujun Yin[2], and Yinghuan Shi[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] School of Electrical and Electronic Engineering, The University of Manchester, UK

Abstract. Designing effective image priors is a key issue to image super-resolution. However, obtaining analytical forms for evaluating the smoothness of the priors is still a difficult and significant task. In this paper, we propose a prior-based method that divides image edges based on the hardness value, replacing the traditional binary classification of edges with a more detailed classification method. Through this partition, we can achieve smoother and better visual effect. Furthermore, we propose a non-uniform refinement approach to effectively improve the speed and reduce the processing time. Experimental results on multiple real world images have demonstrated the advantages of the proposed method over other existing prior methods both in visual effect and processing speed.

Keywords: Super Resolution, Image Prior, Edge Detection.

1 Introduction

Super Resolution (SR) is one of the most significant applications in computer vision and pattern recognition. SR aims to obtain a high resolution (HR) image from a low resolution (LR) image. However, SR is still an ill-posed problem, since LR images are usually the smoothed and downsampled results of corresponding HR images by reducing information. With the development of SR technology, many methods have been proposed in recent decades. Prior-based method is one of the common approaches to solve the SR problem. A prior of natural images is obtained by using technologies such as statistics. A high quality prior, which can capture the regularity of original HR images, has been introduced to constrain the reconstruction problem as well as improve the quality of SR images [1] .

For prior-based methods, there are still several challenges. For example, human vision is sensitive to sawtooth and artifact images generated by interpolation [2]. Moreover, people usually take into account the surrounding condition

* The work is supported by the National Science Foundation of China (Grant Nos.61035003, 61175042, 61021062), the National 973 Program of China (Grant No. 2009CB320702), the 973 Program of Jiangsu, China (Grant No. BK2011005) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 168–177, 2013.

when distinguishing between different colors. Modeling the natural scene to vector image is one of the solutions which can achieve good quality of HR images. However, since the edges of the nature scene are usually continuous curve or oblique irregular lines, it is always too complex and unfeasible to build the vector image model. To address this issue, we try to solve SR problem by analyzing the LCD screens' features. For LCD screens, it is able to achieve unlimited zoom effect in both horizontal and vertical direction because the displayed images are made of pixels and always discrete. But for the curve or oblique edges, the degree of smoothness is determined by the arrangement of pixels. The smaller pixel granularity of the curve line, the smoother the line is. Therefore, we can build a new solution from the characteristics of LCD screens to solve the SR problems.

According to most recent studies, image pixels can be divided into two types, edge or non-edge. This kind of division is not fit for the actual situation, in which the image pixels are usually continuous. Also, the binary division of pixels result in larger pixel granularity and less smoothness. Some other people proposed the interpolation methods [2] such as quadratic or cubic interpolation to generate HR images. Unfortunately, these methods often tend to generate several jag, zoom in the edges of HR images, thus are inefficient to SR of nature images.

A typical example can be referred to Fig.1, where (a) represents the original line; (b) is the one scaled by vector amplification, which has obvious differences compared to the composition of pixels in (a); (c) is the result of linear interpolation amplification, we can found that the corresponding edge becomes fuzzy; (d) amplifies the line directly and the sense of sawtooth of edge is apparent. It is obvious that (b) has the best result. However, as we mentioned in the previous paragraph, this method is too complex and unrealistic. The goal of our method is to achieve the effect of super resolution from (a) to (b) as much as possible.

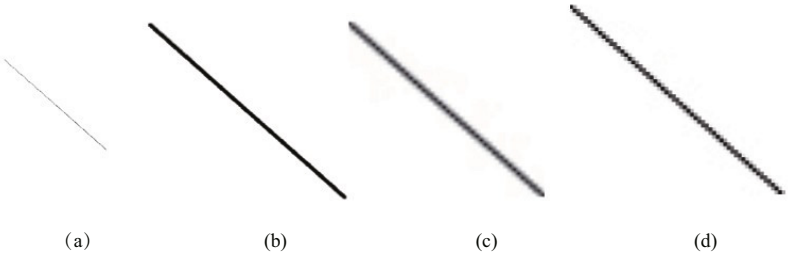


Fig. 1. (a) Original line image, (b) Vector amplification image, (c) Liner interpolation amplification image, (d) Directly amplification image

In this paper, we propose a Visual prior for Digital Image Characteristics (VDIC), which combines the features of digital images and human vision system, and introduces the concept of hardness value, and redefines the image edge by taking into account the condition of the surrounding and the internal objects when we distinguishing between colors. The main contributions of VDIC include (i) a more detailed edge classification and a new edge division method based on the hardness value, which can improve the SR visual effect; (ii) a non-uniform iterative refinement method to speed up the image prior process.

The organization of the paper is as follows. Related works are introduced in Section 2. Section 3 derives the new image prior method VDIC model. Section 4 presents experimental results on multiple real images. Finally, brief conclusions and discussions are provided in Section 5.

2 Related Work

Various image priors have been employed in the recent years. One of the most extensively studied priors is edge prior modeling. Sun *et al.* [1, 3] proposed a general image statistical prior for the gradient field of natural images. This method generalize the general distribution of the image edge gradient which along the vertical direction of image edge.

Based on Geocuts (Geodesics cut) [4] and alpha matting [5] technologies, Dai *et al.* [6–8] found the connection between a soft edge smoothness and a soft cut metric on an image grid, and further proposed the soft edge smoothness prior. In addition, this method combines the soft edge smoothness prior and the alpha matting technique for color SR images, and has been validated for its superior performance.

Besides the edge prior modeling, Shan *et al.* [9] proposed image reconstruction-based gradient density distribution prior. The method is an upsampling method with a feedback-control framework to obtain the image prior.

Edge prior modeling regards the edge as a certain width, while gradient curve segment only focuses on part of the SR problem without taking the texture information of images into account. Moreover, traditional prior methods usually use bicubic interpolation [2, 10] or Iterative Back Projection (IBP) [11] method to iteratively refine the images. Also, it is very time consuming in order to satisfy the convergence condition of IBP. To overcome these challenges, we propose VDIC method to redefine the image edge and introduce a non-uniform iterative refinement method to speed up the image prior process.

3 Methodology

3.1 VDIC Prior

Hardness Value Definition. The traditional methods divided pixels into two types: edge and non-edge, corresponding to outside and inner pixels of objects, respectively. In natural model, the color of an object interior is fairly uniform, while relatively significant differences of color often exist between different objects. We assign each image pixel to a value h ($h \in [0, 1]$). The value of h describes the probability whether the edge is the boundary of objects or the different color inside one object. When the value is closer to 0, it is more likely to be a soft edge which represents inside the object. While the value is close to 1, it is a hard edge which represents edge between two different objects.

To simplify the programming, we always discretize the hardness value: the hardness value h is divided into 64 levels, each level corresponds to a short interval with

gray level as 8. Then, the discrete interval of the hardness value is [0,31]. As for an image edge, if the hardness value h is close to 0, it is considered as a soft edge which shows the pixels on both sides may derive from the same object. Thus, we can adopt interpolation method on the image. The fictitious value can be considered as gradual change. If h is close to 31, it can be called as a hard edge which shows the pixels on both sides may from two different objects. Applying interpolation method is unreasonable because the fictitious value may integrate different pixels of objects. While, pixel filling method can be applied instead of interpolation.

Calculate Hardness Value. In VDIC, Canny edge detection method [12] or optimizing the threshold of Canny can be used to calculate the hardness value. In Canny edge detection, there are two thresholds: caps and collars. In VDIC, lower threshold is $8h$, and upper threshold is $8h + 7$. Canny method detects I_l and gets edge set of different hardness value $g_{h=1}^{31}$, the problem of solving SR problem for I_l now is transferred to solve SR problem for $g_{h=1}^{31}$.

The next step is to calculate required size of SR. Thus, for each edge in $g_{h=1}^{31}$, we can calculate its position of matched edge in I_h. Since bicubic interpolation introduces fictional pixels and the restriction of edge width should be 1, the result E_{h_0} can not be obtained by simply amplifying the E_l. Here, E_{h_0} is the result of interpolated image I_{h_0} by using edge detection method, and E_l is the edge detection result of original image I_l. In other words, the edge pixel of I_h and I_l may not be able to complete alignment with each other. The way to solve this problem is measuring the distance as well as direction between edge pixels, and then choosing the closest matching pixels. Given the point p_i from edge e_i in $g_{h=1}^{31}$, the closest point p_i^h matched with I_h should be:

$$p_i^h = \arg\min_{p \in N(p_i)} \|p - p_i\|_2 + \|\overrightarrow{N}(p) - \overrightarrow{N}(p_i)\|_2, \tag{1}$$

where $N(p_i)$ is $n \times n$ neighborhood of p_i in I_h. Typically, the value of n is always set to 5 or 7. $\overrightarrow{N}$ is gradient vector.

3.2 Non-uniform Iterative Refinement Method

SR method usually optimizes image by means of iterative refinement step. In VDIC, by using the information of edge division mentioned in earlier section, we propose a non-uniform iterative refinement method which optimizes the traditional method.

According to the redistrict of edge, calculating hardness value of edge in each block, constructing mapping matrix $M_{mapping}$ of iteration number, for different image block use different iteration times so that total iteration times can decrease. More iteration time required for hard edge image block and less for soft one. Meanwhile, taking into account of continuity between the connected blocks, we will optimize the statistic result $M_{mapping}$ of initial block so that the iteration times of neighborhood keep stable. Updating the $p(i, j)$ iteration times with proper weight w. After optimization, return the new mapping matrix $\overline{M}_{mapping}$.

Using steepest descent method to optimize object function, then putting the neighborhood group together. Each iteration can be computed as formula (2):

$$I_h^{t+1} = I_h^t + pBP. \tag{2}$$

Where $pBP = ((I^t * G \downarrow -I^l)) \uparrow *G$. β is iteration step, pBP is update function of approximate reverse projection, $\uparrow$ is upsampling operation, while $\downarrow$ is downsampling operation. In the process of approximate reverse projection, introducing reconstruction constrain which based on human visual system. When iteration times are reduced to 0, image blocks stop loop respectively. The mainly procedure of VDIC can be concluded as follows:

Algorithm 1. VDIC Prior Procedure.

Input: LR image I_l and amplification factor m

1: Compute I_h^0 through bicubic interpolation with m, I_h^0 is the initial iteration image of iteration refinement. The I_l and I_h^0 images are partitioned with size n*n and $(m * n) * (m * n)$ respectively;

2: For each block $p(i, j) \in I_l$, calculate hardness value h of edge, determine the required iteration times of IBP, then get the whole image mapping matrix $M_{mapping}$ about IBP iteration times, optimized and return $\overline{M}_{mapping}$;

3: For each block $p(i, j) \in I_h^0$, start loop according to the related iteration times of $\overline{M}_{mapping}$:
(1) Add reconstruction restrain to iteration result based on h and human visual VDIC prior;
(2) Based on non-uniform iterative refinement procedure, repeat downsampling operation and continue iteration refinement.

Output:

4: HR image I_h.

Here, the amplification factor m can be set according to the actual demand. The corresponding iteration time of each image blocks can be computed by $\overline{M}_{mapping}$.

4 Experiments and Results

4.1 Datasets and Experiment Settings

We tested our approach on a set of generic images. The input low resolution image is produced by blurring and downsampling the high resolution image. Our experimental results are shown in Figs. 2-6, all with a magnification factor of 3. We conducted four experiments and compared with several state-of-the-art prior methods [9, 13–15]. For each experiment, we use RMS and ERMS [16] to quantify and compare the performance of each super resolution method. The smaller the value, the less disparity between SR images with original image. RMS tends to measure global performance while ERMS focuses on the edge part. Meanwhile, we compute

the iteration time of different methods to reflect the speed of the SR process. The run time of this algorithm is 20-60 seconds per image on a Intel(R) Core(TM) i5 3.20GHz CPU for all the images in our experiments.

4.2 Compare with Other Traditional Prior Methods

We compare our VDIC prior with other traditional image prior methods. In each experiment result, (a) is original image, (b) is result of Fattal [13], (c) is result of Freeman[14], (d) is result of Qi [9], (e) is the result of Daniel[15] and (f) is our VDIC prior. The experiment results show that our approach can obtain higher quality of HR images and less iteration times required for each SR problem.

In Fig.2 the resolution of image is 128×128. The edge region of (c) in Fig.2 is lack of smooth and appears a few artifacts. While, VDIC prior result shows in (f) has very high smoothness, especially for the description of the eyebrows and lips. In Fig.3, resolution of (a) is 244×200. We can see that in (f) the characters on chip are much clearer than other results. Moreover, VDIC method can achieve stronger contrast. For example, by comparing performance in (d) and (f) in Fig.3. In Fig.4, resolution of (a) is 68×83. The color of this image is very simple, so we will pay more attention to performance of the texture. The texture of (b)(c)(d)(f) doesn't restore in its true sense. Result in (d) has good performance of texture however poor performance in the restore of edge region. The texture of (b) is also a little poor. While, VDIC method result shows in (f) can restore texture very well as well as the edge region. For the fourth experiment, the image we choose in Fig.5(a) the resolution is 161×241. This image is complex, higher demand to restore fur of koala. The restore of the tree texture in (e) is not very good. VDIC method has the best edge detection while much better texture performance than (b)(c)(d)methods.

Table.1 and Table.2 show the RMS and ERMS result of SR with different methods and VDIC has much higher performance than others. We can see that the RMS result of VDIC is the smallest in first and third experiment, and the second smallest in second and fourth experiment. Meanwhile, about ERMS result, VDIC is the smallest at most time. It means that the VDIC outperforms the other approaches both around the edge regions and whole images. Meanwhile, we can see in Fig.9 that by using non-uniform iteration refinement method of VDIC can accelerate the speed of iteration and minimize the iteration time.

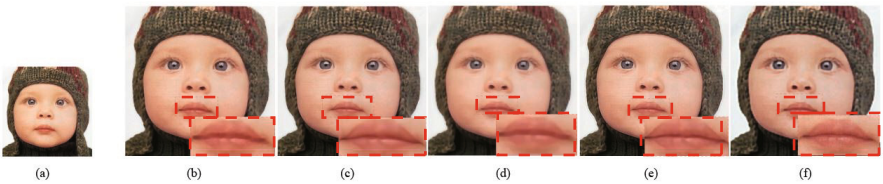


Fig. 2. The "Child" image magnified 3X using (b) Fattal [13], (c) Freeman [14], (d) Qi [9], (e) Daniel [15] and (f) our approach. Very high smoothness, especially for the description of the eyebrows and lips in our result.

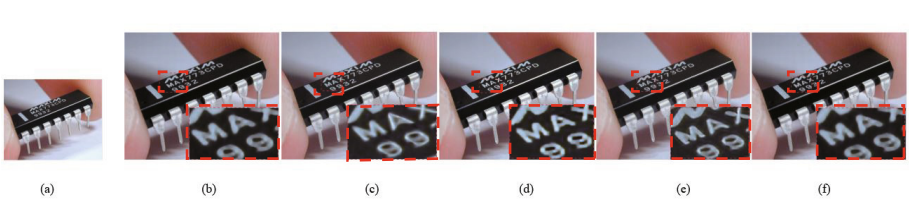


Fig. 3. The "Chip" image magnified 3X using (b) Fattal [13], (c) Freeman [14], (d) Qi [9], (e) Daniel [15] and (f) our approach. The characters on chip are much clearer than other results.

Fig. 4. The "Statue" image magnified 3X using (b) Fattal [13], (c) Freeman [14], (d) Qi [9], (e) Daniel [15] and (f) our approach. Better texture restore performance in our result.

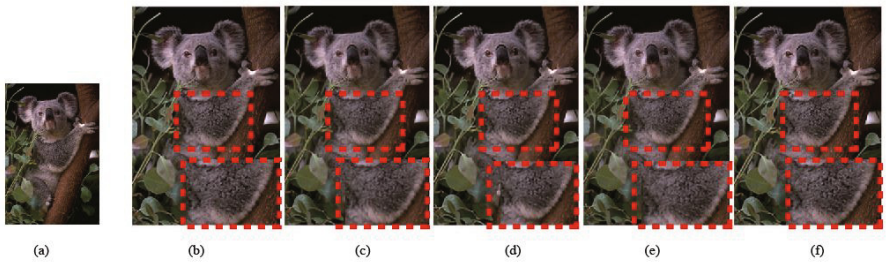


Fig. 5. The "Koala" image magnified 3X using (b) Fattal [13], (c) Freeman [14], (d) Qi [9], (e) Daniel [15] and (f) our approach. Better texture restore performance in our result.

Table 1. RMS pixel error for different approaches

Test Image	Fattal	Freeman	Shan	Daniel	VDIC
Child	14.0	10.1	9.6	10.54	9.2
Chip	11.3	7.8	7.4	8.4	7.8
Statue	7.4	8.3	7.8	7.1	6.0
Koala	32.1	42.0	38.0	59.1	35.8

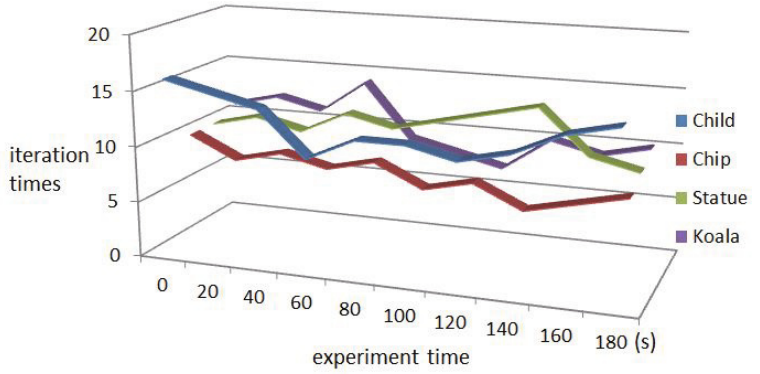


Fig. 6. VDIC iteration time sketch. Each line represents the iteration of a pair of images. The horizontal axis represents the SR experiment time and time units are seconds(s). The vertical axis represents the iteration time of image blocks. This figure shows that our VDIC method has high iteration speed.

4.3 Super-Resolution Magnification Factor

Super resolution magnification factor (MF) reflects the magnification from LR images to HR images. Here, we choose different parameter values from 1.0 to 4.0 to observe the super resolution results. Fig.7 is the outcome which shows that lower SR magnification factor corresponding to better performance. Meanwhile, the RMS results in Fig. 8 also explain the influence of SR magnification factor that with the increase of SR magnification factor the RMS values tend to increase. As for the iteration time which is showed in Fig.9, higher SR magnification factor leads to higher iteration time and lower speed of SR process since with the increase of SR magnification factor, more pixels need to be calculated during the process of reconstruction.

Table 2. ERMS pixel error of different approaches

Test Image	Fattal	Freeman	Shan	Daniel	VDIC
Child	18.1	14.9	13.82	15.4	12.9
Chip	14.5	10.2	9.3	11.6	10.2
Statue	9.3	11.4	10.2	9.1	7.4
Koala	38.4	45.4	40.9	64.7	40.4

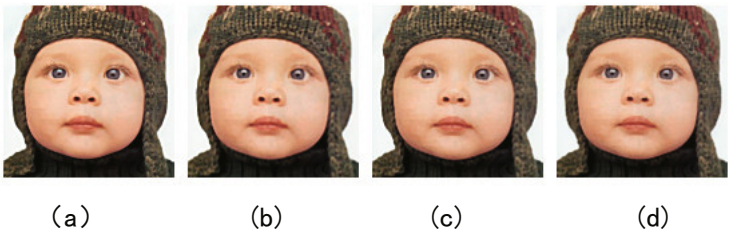


Fig. 7. Results with different super resolution ratios. (a)SR magnification factor is 1.0 (b)SR magnification factor is 2.0 (c)SR magnification factor is 3.0 (d)SR magnification factor is 4.0.

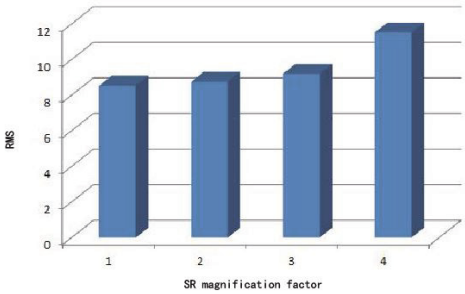


Fig. 8. RMS result for different SR magnification factor change from 1.0 to 4.0. The horizontal axis represents SR magnification factor and vertical axis is RMS value. With the increase of the magnification factor, RMS value increase gradually.

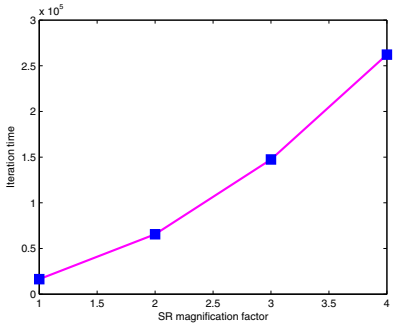


Fig. 9. Iteration time change with different SR magnification factor. The horizontal axis represents SR magnification factor and vertical axis is iteration time. As the magnification factor increased from 1.0 to 4.0, the iteration number increased rapidly.

5 Conclusions

In this paper we have proposed a novel image prior method termed VDIC. Experimental results on several real world images validated the effectiveness of the proposed VDIC model. VDIC examines the conventional edges the same way as the interior pixels of an image with hardness values. The function of the hardness value

is a continuous function that has two obvious advantages: First, according to different hardness value, different kind of edges can be identified with different strategies. Higher hardness values use the method while lower values mainly use the interpolation approach. Second, we can use non-uniform refinement approach based on the hardness value in the process of iteration refinement, and this approach can reduce the iteration numbers so to improve the speed significantly. Future work includes to improve the performance of our approach in texture detection. The approach can be extended to other image applications to speed up the processing time.

References

1. Sun, J., Xu, Z., Shum, H.Y.: Image super-resolution using gradient profile prior. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
2. Hou, H., Andrews, H.: Cubic splines for image interpolation and digital filtering. IEEE Transactions on Acoustics, Speech and Signal Processing 26(6), 508–517 (1978)
3. Sun, J., Xu, Z., Shum, H.Y.: Gradient profile prior and its applications in image super-resolution and enhancement. IEEE Transactions on Image Processing 20(6), 1529–1542 (2011)
4. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proceedings of the Ninth IEEE International Conference on Computer Vision 2003, pp. 26–33. IEEE (2003)
5. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 228–242 (2008)
6. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y.: Soft edge smoothness prior for alpha channel super resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
7. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y., Katsaggelos, A.K.: Softcuts: a soft edge smoothness prior for color image super-resolution. IEEE Transactions on Image Processing 18(5), 969–981 (2009)
8. Dai, S.: Conquering image imperfectness by priors with applications to image recovery (2009)
9. Shan, Q., Li, Z., Jia, J., Tang, C.K.: Fast image/video upsampling. ACM Transactions on Graphics (TOG) 27, 153 (2008)
10. Keys, R.: Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech and Signal Processing 29(6), 1153–1160 (1981)
11. Irani, M., Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency. Journal of Visual Communication and Image Representation 4(4), 1993–1912 (1993)
12. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (6), 679–698 (1986)
13. Fattal, R.: Image upsampling via imposed edge statistics. ACM Transactions on Graphics (TOG) 26(3), 95 (2007)
14. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM Transactions on Graphics (TOG) 30(2), 12 (2011)
15. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: IEEE 12th International Conference on Computer Vision 2009, pp. 349–356. IEEE (2009)
16. Sun, J., Zheng, N.N., Tao, H., Shum, H.Y.: Image hallucination with primal sketch priors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003, vol. 2, pp. II–729. IEEE (2003)

Deep Learning on Natural Viewing Behaviors to Differentiate Children with Fetal Alcohol Spectrum Disorder

Po-He Tseng[1], Angelina Paolozza[2], Douglas P. Munoz[2], James N. Reynolds[2], and Laurent Itti[1]

[1] Department of Computer Science, University of Southern California, USA
[2] Centre for Neuroscience Studies, Queen's University, Canada

Abstract. Computational models of visual attention have attracted strong interest by accurately predicting how humans deploy attention. However, little research has utilized these models to detect clinical populations whose attention control has been affected by neurological disorders. We designed a framework to decypher disorders from the joint analysis of video and patients' natural eye movement behaviors (watch television for 5 minutes). We employ convolutional deep neural networks to extract visual features in real-time at the point of gaze, followed by SVM and Adaboost to classify typically developing children vs. children with fetal alcohol spectrum disorder (FASD), who exhibit impaired attentional control. The classifier achieved 74.1% accuracy (ROC: 0.82). Our results demonstrate that there is substantial information about attentional control in even very short recordings of natural viewing behavior. Our new method could lead to high-throughput, low-cost screening tools for identifying individuals with deficits in attentional control.

Keywords: deep learning, sparse representation, eye movement, natural scenes, clinical populations.

1 Introduction

Attention enables us to interact with complex environments by selecting relevant information to be processed in the brain. Eye movements have served as a probe for visual attention in a wide range of conditions, because both are strongly coupled in our everyday life [8]. A large brain network is recruited by attention, which makes attention control susceptible to neurological disorders [1]. Thus, psychologists have designed eye-tracking experiments with simple artificial stimuli to characterize the influence of these disorders on eye movement metrics and to assist diagnosis [8].

In the past decades, computational models of visual attention have shown significant ability to predict human eye movement while people view complex natural scenes and perform complex tasks. The success of these attention models, combined with eye tracking data, allows us to assess how neurological disorders

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 178–185, 2013.

impact attention under a more natural scenario, compared to traditional psychophysical experiments with artificial stimuli. Previous research has utilized attention models to assess autism [2] and agnosia [11,3], however, without much success (null or inconsistent results). This is possibly due to the complexity of natural scenes and natural viewing eye traces, especially when the differences between patients and controls are subtle.

In this study, we propose a novel method that allows us to use only 5 minutes of natural viewing eye traces to identify children with FASD. FASD is caused by prenatal exposure to alcohol, which impacts the brain globally and impairs attentional control. Our new method computes "attention traces" by collecting the predictions of attention models (saliency values) along recorded human eye movement traces. To represent these complex attention traces, we utilize a convolutional deep neural network [9] to learn their sparse representation in an unsupervised manner, and to discover the features that best capture variations in attention traces; we then differentiate children with FASD from typically developing children by classifying these features over time.

2 Methods

2.1 Paradigm Overview

Participants' eye movements were recorded while they watched five 1-minute videos, and they were instructed to "watch and enjoy the clips". These clips were composed of content-unrelated clip snippets of 2-4 seconds each (a few were 7-8 seconds). This design attempted to reset the observers' attention status with each new snippet (Fig. 1A). For each video frame, Itti's visual attention model was used to compute a saliency map that predicts visually conspicuous locations from low-level visual attributes of the video frame. The normalized map values were extracted along the eye trace (attention trace). To extract features from the attention traces, a convolutional deep neural network was utilized to discover 256 bases from the attention traces of control young adults. Next, we applied these bases on the child data and obtained a sparse representation of their attention traces. This representation was input to a L1-regularized logistic SVM classifier after initial filter feature selection. We built a weak classifier from each clip snippet independently, and used Adaboost to construct a strong classifier for the final prediction.

2.2 Computational Model of Visual Attention

To evaluate an observer's bottom-up attention, we used the popular Itti *et al.* saliency model [7,6,10] to identify visually salient locations that may capture attention in natural videos. Itti's model was biologically inspired to mimic early visual processing in humans.

To estimate visual salience in a scene, the saliency model (Fig. 1B) first applied a linear filter for each of several visual attributes (e.g., color, intensity)

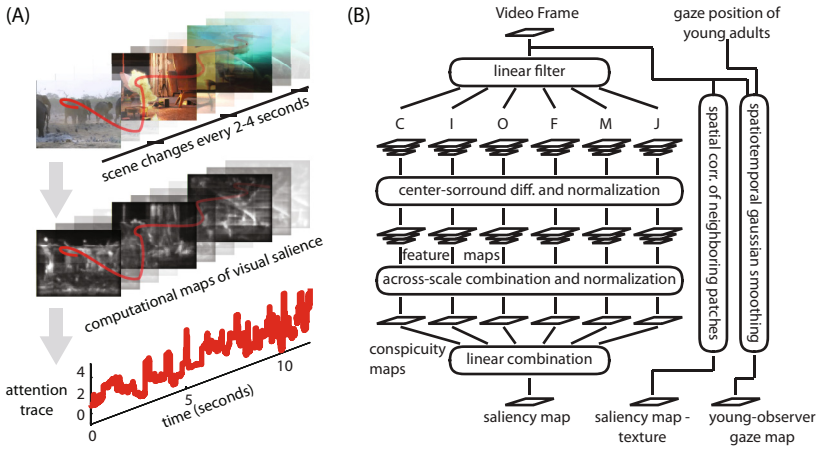


Fig. 1. Evaluating attention deployment. (A) To extract an attention trace, observers' eye movements were recorded (red curve) during free viewing of videos of natural scenes. For each corresponding video frame, ten different types of saliency maps were generated. Brighter intensity in a map indicates locations of the video frame with stronger feature contrast. The normalized map values at every gaze location were extracted to yield the attention trace. (B) Architecture of Itti's computational model of visual attention, extended here to contain many more feature channels than the original implementation. C, color; I, intensity; O, orientation; F, flicker; M, motion; J, line junction.

on a video frame, at several scales to generate multi-scale (i.e., fine to coarse) filtered maps. Fine filtered maps were then subtracted from coarse filtered maps to simulate center-surround operations in human vision and to produce feature maps. Next, these multi-scale feature maps were normalized and combined together to generate conspicuity maps in a manner that favors feature maps with sparse peak responses. Finally, conspicuity maps of different features were linearly summed together to form a saliency map, if multiple visual attributes were evaluated together. Equations of these linear filters were described in [7,6,10].

Ten types of maps used in this study provided information on attention control. Nine saliency maps of different visual attributes estimate bottom-up, stimulus-driven attention, and one top-down map, generated by instantaneous gaze positions of 19 normative young adults, provides information on voluntary, top-down attention control in addition to bottom-up attention. Out of the 9 saliency maps, 7 were individual visual attributes (color, intensity, orientation, flicker, motion, line junction, and texture [10,12]), and 2 were combination of multiple visual attributes (CIOFM and CIOFMJ; Fig. 1B for abbreviations). For each map, we quantified the agreement between eye movement traces and the map, and such agreement over time yielded attention traces (see Supplementary Methods[1]). With the ten maps, ten attention traces were obtained. While eye

[1] Available at `http://ilab.usc.edu/publications/doc/tseng_etal13idealsupp.pdf`

movements were recorded at 500Hz, attention traces were resampled to 125Hz to reduce the computational load. Next, the 10 attention traces were whitened by principal component analysis (PCA), and the first 7 principal components were kept to maintain 97.5% of the information.

2.3 Convolutional Deep Neural Network

We chose a two-layer convolutional deep neural network (CDNN) [9] to discover the sparse representation of the attention traces. The sparse representation had several advantages in representing input signals, and the one which benefited classification the most was denoising, because signals and noise were likely to be separated by different components of the representation. Each layer of the CDNN was a topographic independent component analysis (TICA) network (Fig. 2) [5]. TICA discovers components (bases) from unlabelled data in a way similar to independent component analysis (ICA). However, TICA features a loosened independence constraint between neighboring components so that similar components were next to each other, which gave rise to a more stable set of independent components [5].

Our TICA network was composed of three components: input, simple, and pooling nodes (Fig. 2A). Given an input pattern $x^{(t)}$, each pooling node was activated as $p_i(x^{(t)}; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} (\sum_{j=1}^{n} W_{kj} x_j^{(t)})^2}$, where n was the local receptive field size (size of the input) and m was the number of simple units. The weights $V \in \mathbb{R}^{m \times m}$ between the simple and pooling nodes were fixed ($V_{ij} = 0\, or\, 1$) to represent to topographical structure. The weights $W \in \mathbb{R}^{m \times n}$ between the input and simple nodes were the sparse bases to be learned by TICA. The TICA learned the weights W by solving

$$\arg\min_{W} \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^{(t)}; W, V), \quad subject\ to \quad WW^T = I \tag{1}$$

where input patterns $\{x^{(t)}\}_{t=1}^{T}$ were whitened attention traces. This objective function minimized the sum of activities from all pooling nodes to reduce redundancies between them, and the constraint $WW^T = I$ ensured sparseness between bases. This objective function corresponded to classic ICA if each pooling node p_i was connected to exactly one simple node h_i.

The sparse representation of children's attention traces was simply obtained by propagating their whitened traces through the CDNN pretrained from the young adult controls. The children's attention traces were whitened by the same principal components that young adult controls used. These whitened traces then propagated through the CDNN composed of two TICA networks. For each TICA network, the output of each basis along the whole attention trace was called a "trace-map", and all the trace-maps together should cover the entire attention trace. We concatenated all the trace-maps into one large vector that sparsely represented the saliency traces and would be the features used for classification (see Supplementary Methods for detail).

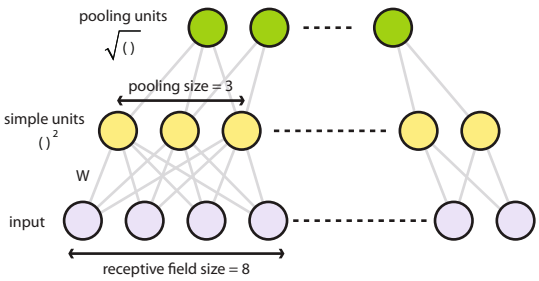


Fig. 2. One TICA layer (itself consisting of 3 layers) of the convolutional deep neural network, where the simple units are the squared weighted sum of the receptive field, and the pooling units compute the square root of the sum of adjacent simple units

2.4 Classification Procedures

A weak support vector machine (SVM) classifier was built per clip snippet, and a strong classifier was constructed by Adaboost [4], from the weak classifiers that performed better than the chance level ($\frac{max(n_i)}{\sum n_i}$, n_i: number of participants of group i). For each weak classifier, the feature dimension (all the trace-maps) was much higher than the number of training samples, e.g., a clip snippet of 3.129 seconds yields 17,408 features. To reduce feature dimensions, a combination of filter and wrapper methods was used in selecting features that discriminated patients from controls in the training data set. For the filter method, we used a two-tail t-test to filter features whose p-value ≤ 0.05 after Bonferroni correction within each trace-map. Once the filter method selected a set of features, an L1-regularized logistic support vector machine (SVM) was used to perform another feature selection and classification simultaneously, and resulted in our weak classifier, which used 33 out of the 17,408 features of this 3.129 second clip snippet. A 5-fold cross validation was used on the training dataset to determine the cost value (10, 1, 0.1, 0.0001) of the SVM.

Within the training dataset, we performed a leave-one-out cross validation (LOOCV) to estimate the generalizability of each weak classifier to different training data, and to estimate the consistency of predictions on test data. The LOOCV left one participant per group out. A weak classifier would be recruited in the Adaboost strong classifier if it met 2 criteria: (1) the weak classifier performed above chance level in the training dataset (generalizability), and (2) the testing data were predicted consistently during the LOOCV. To estimate the consistency, we took the median of the predicted class probability throughout the LOOCV. If the median was higher than 0.7 or below 0.3, then the weak classifier was considered consistent. Moreover, the median predicted class label was the prediction that served as the input of the strong Adaboost classifier.

LOOCV was also used to test the performance of the strong classifier. We first left one participant out. With the remaining participants, we performed the filter feature selection, trained the L1-regularized SVM, selected the weak classifiers based on their generalizability, and trained the adaboost classifier. Finally, the learned classifier was used on the participant who was left out at the beginning. The LOOCV was repeated 10 times with different fold structures. Note that we did not balance the number of participants in each group because our data were only slightly unbalanced (53 controls vs. 47 patients).

3 Experimental Evaluation

3.1 Data

The results reported in this study included 53 control children (11.11±3.31 yr) and 47 children with FASD (12.21±3.32 yr). An additional nineteen young adult controls (20.74±1.33 yr) were recruited to collect data to train the sparse representation of the attention trace by CDNN. The participants watched high-definition (1280×1024 pixels) continuous natural scene videos whose content changed every 2-4 seconds (61 snippets) or 7-8 seconds (9 snippets). Each video (about 1 minute) included 13 to 15 snippets. Participants were instructed to "watch and enjoy the clips", and their right eye was tracked at 500Hz when they watched the videos (see Supplementary Experiment for detail).

3.2 Results

We compared the classification performance of the classifier using either (1) the gaze positions in screen coordinates, (2) the 10 raw attention traces, or (3) the attention traces in sparse representations. All 3 test cases used a t-test filter, L1-regularized SVM, and the adaboost classifier. They only differed in the representation of the eye traces. These classifiers achieved accuracies of 49.3%, 53.4%, and 74.1% respectively (Fig. 3A). Wilcoxon signed-rank tests showed that the classifier using the sparsely represented attention traces outperformed the classifiers using gaze positions and raw attention traces ($p<0.01$). Furthermore, the classifier using raw attention traces performed slightly better than that using gaze positions ($p=0.04$). The confusion matrix of the classifier with sparsely represented attention traces (Fig. 3B) showed precision 73.9%, recall (sensitivity) 69.4%, and specificity 78.3% on average.

To perform receiver operating characteristic (ROC) analysis, decision values of the Adaboost classifier were normalized and converted to probabilities based on Bayes rules. The ROC analysis showed that the classifier with sparse attention traces (area under the curve, AUC=0.82) outperformed the classifiers with gaze positions (AUC=0.49) and with raw attention traces (AUC=0.56) ($p<0.01$). Moreover, the classifier with raw attention traces performed better than the classifier using gaze positions ($p=0.02$).

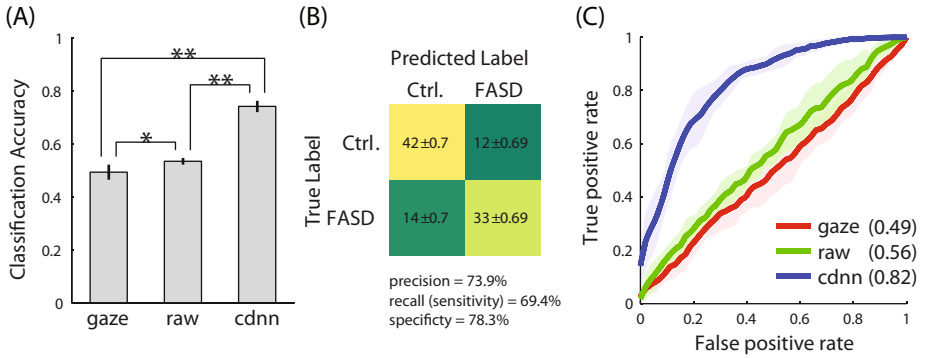


Fig. 3. (A) Classification accuracies of the classifiers using the gaze positions (*gaze*), the raw attention traces (*raw*), and the attention traces in sparse representation learned by convolutional deep neural network (*cdnn*). *, $p<0.05$; **, $p<0.01$ (B) Confusion matrix of the classifier using the sparsely represented attention traces. (c) ROC analysis. The numbers inside parentheses are area under the curve for each classifier.

3.3 Comparisons

Diagnosing FASD currently is a complex and time-consuming process. Several studies have attempted to assist diagnosis by building classification rules that are based on objective measures on how FASD influences children's brain structure and behaviors. However, most of such studies used the same dataset to train and test their classifier, and some studies used the same dataset to select features and to train the classifier (double-dipping) (see Supplementary Comparisons). Therefore, their classification performance may be overestimated.

One fair comparison is the study done by Tseng et al. [12]. Similar to our study, they investigated 15 minutes of natural viewing behavior combined with Itti's computational model of visual attention to classify children with FASD from controls. They reported 79.2% classification accuracy by leave-one-out cross validation, which was comparable to our results. However, their work was based on only one type of eye movement, saccades, which occured only $\sim$1.5 times per second on average for children. The low saccade frequency may limit their method to be further shortened, because small numbers of saccades may not provide reliable statistics to describe attention deployment. In fact, we used their algorithm (provided by the authors) on our 5 minutes of data, and the classification accuracy was 54.1%, which was lower than ours (74.1%).

4 Conclusions and Future Work

This study demonstrates that substantial information about the state of one's attention control exists in one's natural viewing behavior. By combining computational models of visual attention and convolutional deep neural networks, such

information can be successfully decoded. With such a short experiment and the low cost of eye trackers, this method shows potential to be used as a screening tool that can be broadly deployed (unlike, e.g., MRI).

This study serves as the first step in decoding attention control status from natural behavior, and several extensions of this work can be explored. For example, video stimuli can be designed and mined to specifically discriminate different populations or disorders. Different representations of the attention traces can be investigated, and each attention trace for each feature can be separately classified to give rise to a multi-dimensional diagnostic profile (biometric signature). Correlating the classifier to standard behavioral tests and functional/structural imaging data can advance our understanding of what the discriminative information is which the classifier learned. The underlying generating processes of attentional control can be modelled to better understand attentional control. In summary, this study opens up a new research area that is exciting to explore.

References

1. Corbetta, M., Patel, G., Shulman, G.L.: The reorienting system of the human brain: from environment to theory of mind. Neuron 58(3), 306–324 (2008)
2. Fletcher-Watson, S., Leekam, S.R., Benson, V., Frank, M.C., Findlay, J.M.: Eye-movements reveal attention to social information in autism spectrum disorder. Neuropsychologia 47(1), 248–257 (2009)
3. Foulsham, T., Barton, J.J.S., Kingstone, A., Dewhurst, R., Underwood, G.: Modeling eye movements in visual agnosia with a saliency map approach: bottom-up guidance or top-down strategy? Neural Networks 24(6), 665–677 (2011)
4. Freund, Y., Schapire, R.: A desicion-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory 904(1), 23–37 (1995)
5. Hyvärinen, A., Hoyer, P.O., Inki, M.: Topographic independent component analysis. Neural Computation 13(7), 1527–1558 (2001)
6. Itti, L., Dhavale, N., Pighin, F.: Realistic Avatar Eye and Head Animation Using a Neurobiological Model of Visual Attention. In: Bosacchi, B., Fogel, D.B., Bezdek, J.C. (eds.) Proc. of the SPIE 48th Annual International Symposium on Optical Science and Technology, vol. 5200, pp. 64–78. SPIE Press, Bellingham (2003)
7. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11), 1254–1259 (1998)
8. Karatekin, C.: Eye tracking studies of normative and atypical development. Developmental Review 27(3), 283–348 (2007)
9. Le, Q.Q., Ngiam, J., Chen, Z., Hao Chia, D.J., Koh, P.W., Ng, A.Y., Chia, D.: Tiled convolutional neural networks. In: Lafferty, J., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 23, pp. 1279–1287 (2010)
10. Li, Z., Itti, L.: Saliency and Gist Features for Target Detection in Satellite Images. IEEE Transactions on Image Processing (December 2010)
11. Mannan, S.K., Kennard, C., Husain, M.: The role of visual salience in directing eye movements in visual object agnosia. Current Biology 19(6), R247–R248 (2009)
12. Tseng, P.H., Cameron, I.G.M., Pari, G., Reynolds, J.N., Munoz, D.P., Itti, L.: High-throughput classification of clinical populations from natural viewing eye movements. Journal of Neurology (August 2012)

Prevailing Trends Detection of Public Opinions Based on Tianya Forum

Lina Cao and Xijin Tang

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China (100190)
{caolina,xjtang}@amss.ac.cn

Abstract. How to catch both the central topics and the trend of topics over the on-line discussions are not only of theoretical importance for scientific research, but also of practical importance for social management in current China. In social management perspective, making intervention toward crisis timely and precisely depends on the right image or perception of public opinions toward the crisis. In our research, topic modeling is applied to explore the changing topics of new posts collected from Tianya Zatan Board of Tianya Club. Those online data reflect the community opinions toward social problems.

Keywords: topics detection, dynamic topic models, Tianya Club.

1 Introduction

In current China, more and more people treat social media as a place to express their opinions about almost everything openly and freely. Toward the social events, besides the traditional media, on-line discussions show fresh, diverse, and evolving opinions which may drive big changes toward public life, and exert a broad and far-reaching influence on community, traditional media and government decisions. It is not only of great value to detect the hazards that people concern [1] and monitor the evolution of hot topics, but also helpful to make reasonable and timely interventions for better management.

In our research, we focus on topics mining over textual data on Tianya Zatan board [2], the 2nd largest board within Chinese Tianya Club. The Forum provides a suitable data source to research the occurrence and evolution of topics about daily lives, social unfair, corruption, phenomena of society, etc.

We use topic models to study the fluctuation of topics in a period and analyze the details of words evolution in the topics in this paper. The paper is organized as follows. Firstly, the related work is reviewed and the dynamic topic model (DTM) is briefly introduced. Then, processing of Tianya Forum posts is addressed, and the results of the data analysis are given. Last, conclusions are given.

2 How to Extract Topics from Textual Opinions

In reality, getting a rough image of a concerned issue is quite important for wicked problem solving, especially the online governance in China. In the past, we did some

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 186–193, 2013.

researches using qualitative meta-synthesis technologies, denoted as iView and CorMap, to extract group's opinions and compared the results of different technologies [3]. Those methods are applied to acquire a structure or systemic vision from a group of textual data, draw a scenario using different clustering based on different modeling toward group arguments and extract concepts from clusters of arguments from multi-perspectives [4]. Those studies are exploratory analysis. At those studies, we regard those concepts are topics, central ideas, etc.

Another trend is topic modeling by computer scientists who are on machine learning. Topic modeling has become a powerful tool for "extracting surprisingly interpretable and useful structure without any explicit 'understanding' of the language by computer" [5]. To identify and track dynamic topics from time-stamped textual data, topic models also have been applied. According to discrete or continuous time, topic models can be generally divided into three categories. The first is basic topic model, Latent Dirichlet Allocation (LDA) model, which treats documents as *bags of words* generated by one or more topics [6]. This model is applied to the whole documents to induce topics sets and then classify the subsets according to the time of documents [7]. The second kind, such as Dynamic topic models (DTM) [8] and Online LDA (OLDA) model [9], marks documents according to the discrete time before the generative process. The third kind includes Topics over Time (TOT) model which captures both word co-occurrences and localization in continuous time [10] and the continuous dynamic topic model (cDTM) which replaces the discrete state space model of the DTM with its continuous generalization, Brownian motion [11].

Those models have been successfully applied to explore and predict the underlying structure of various data, such as research papers [9-11], newswire articles [10], personal emails [11], and movie synopsis [12]. Next, we apply DTM to Chinese forum data.

3 Topics Detection from Tianya Club

Tianya Club has been once the biggest Internet forum in China with approximately 91% visitors from China mainland. It has become a comprehensive virtual community that combines online forums, social media and blogging. Among a variety of boards, Tianya Zatan, the 2nd largest board within Tianya Club, is a specific board towards the phenomena of society. Hot events can always been widely and deeply discussed in that board, as illustrated by those posts.

In order to study the social stability in current China society, we start to collect data from Tianya Zatan board and several other relevant boards since October of 2010. Now there are about 2,000 new posts published and 4,000 plus posts updated every day. The accumulative data are about 3 million posts. In this paper, we test topic modeling to the posts of this board.

3.1 Forum Data Processing

We select new posts published during December of 2011 to March of 2012. Data processing excludes posts with empty contents which may be removed due to

censership, includes posts with contents. After segmentation by ICTCLAS[1], Baidu hot search news words are adopted as reserved words [13] and nouns and gerunds are selected for analysis. Then, terms that occur fewer than 50 times in corpus and in fewer than 10 posts are removed. The 4-month data are cleaned respectively as the data source. Tabel1 lists the number of posts, corpus and dictionary in each month.

Table 1. The statistics of data sets

Time span / Data statistics	Dec. 2011	Jan. 2012	Feb. 2012	Mar. 2012
New posts #	12,155	12,032	20,124	37,549
Corpus #	14 million	12 million	23 million	45 million
Dictionary #	4,541	3,973	6,091	9,516

3.2 Topic Modeling Experiments

In DTM, time slice is used to divide the data. How to decide the granularity of time is a quite important but confusing task. We attempt a series of experiments with 1-day interval, 3-day interval and 7-day interval running through DTM package[2] and analyze results using R. It seems that it is not suitable to train our data with too long interval. Unlike the research articles that the new papers definitely are based on the existing research with good consistency, the BBS posts reflect daily life then long time interval models only catch those topics last longer and "ignore" some suddenly happened soon disappeared hot topics. Another problem is to define the appropriate span of time. DTM assumes that the number of topics is fixed and the topics extend the span of time. On the Web, topics are updated quickly and many last only several days, such as topics on the festival. Thus, long time span may not be appropriate.

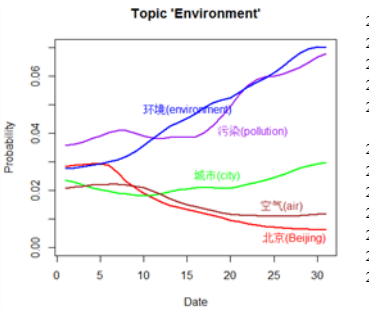
Then, four models with four data sets are trained respectively with 60 topics and 1-day time slice.

3.3 Words Variety

At the topic level, each topic is now a sequence of distributions over words according to the posterior inference. In practice, different terms may be used in different periods when people discussing. Thus, the distributions over words reflect the changing of the key point of the discussion. We select a topic with words about "environment" for detail analysis. Fig.1 shows the varying top several words distributions in this topic through different perspectives. Fig.1 (a) shows the trends of words along the whole December. The y-axis is the probability distribution of words under the respective topic. Fig.1 (c) shows top 5 words within the topics in several time stamps. 10 posts which exhibit these topics are selected and their thread titles are listed by time in Fig.1 (b).

[1] ICTCLAS is a widely used Chinese segmentation program. The website is `http://www.ictclas.org/`

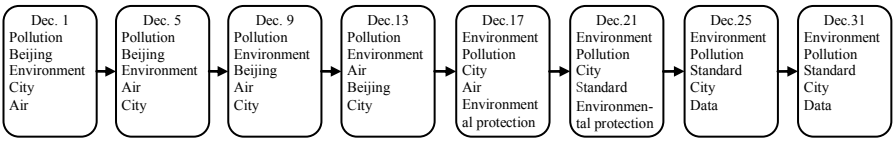
[2] The DTM code package can be downloaded from `http://www.cs.princeton.edu/~blei/`



(a)

2011-12-02 北京 PM2.5 高的快让人窒息了？有木有！
2011-12-06 空气污染与肺癌的相关报道(综合)
2011-12-07 空气环保只能看天意
2011-12-08 世卫城市空气质量排名：北京杭州均列千名之外
2011-12-10 全球 1100 个城市空气质量排名涉及 32 个中国城市，海口排名最高，814 位
2011-12-14 我们的环境好转了吗？
2011-12-15 北京"蓝天"不蓝的重大发现
2011-12-19 中国机动车年排污逾 5000 万吨 成大气污染重要来源
2011-12-20 机动车排放之严重污染恐仍将继续
2011-12-26 "蓝天梦"如此完成：挤出来仅是水分？
2011-12-27 关注 PM2.5 履行企业社会责任
2011-12-31 是轻度污染还是危险？

(b)



(c)

Fig. 1. Results of word trend analysis on topics "environment"

We go further to analyze the topic variability in the time series from a level of corpus.

3.4 Topics Evolution Analysis

According to the posterior inference of the topics distribution, the mixture of topics in each post is obtained. In order to obtain an intuitional sense about the prevailing trends of topics, a statistical variable is designed to depict the evolution.

Algorithm. Let $\theta_{d,t}$ denotes the topic distribution for post *d* in time *t*. In our work, an *average* θ_t (denotes as $\overline{\theta_t}$) is calculated for each post on each day to depict the average hot degree of topics per-day. The volatility of $\overline{\theta_t}$ reflects the changing of public foci on topics. The *average* θ_t is calculated by following steps:

Step 1: Sort all posts according the time stamp and count the number of posts M_t in each time slice;

Step 2: At time *t*, sum up the proportions of posts for each topic, denotes that $\theta_t = \Sigma_d \theta_{d,t}$;

Step 3: For each topic, θ_t divided by the number of posts, M_t, denotes that mean $\overline{\theta_t} = \frac{1}{M_t}\sum_d \theta_{d,t}$;

Step 4: For all $t = 1,\ldots,T$, repeat *Step 2* and *Step 3*.

The calculation results of the *average* θ_t of four months are drawn in Fig.2 in sequence. In each month, the y-axis is 30 topics which are picked with high value of θ_t and ordered by the value.

Result Explanations. We analyze the results which are calculated by the algorithm month by month. In December of 2011, topics such as *general mood of society*, *people's life* and *marriage* were always being highly referred. The bad news such as milk poor quality and school bus accident caused the topic *food safety* and *school bus* growth due to their importance. Some topics were related to specific days such as *Christmas* and *railway*. People talked about *railway* because the New Year was coming and many people might go travel by train, then a small rise of probability was around December 21, the first day to allow on-line ticket order.

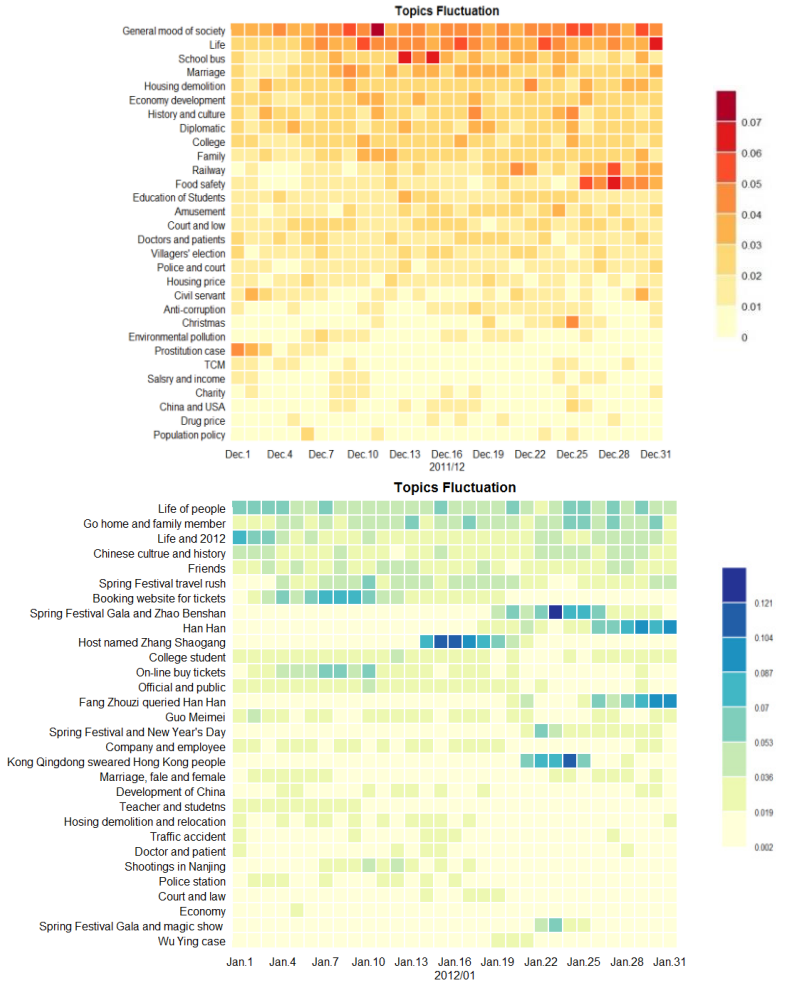


Fig. 2. The horizontal axis represents time whereas the vertical axis represents topics. The cell represents the value of *average* θ_t in each day and its color indicates the interval of the value.

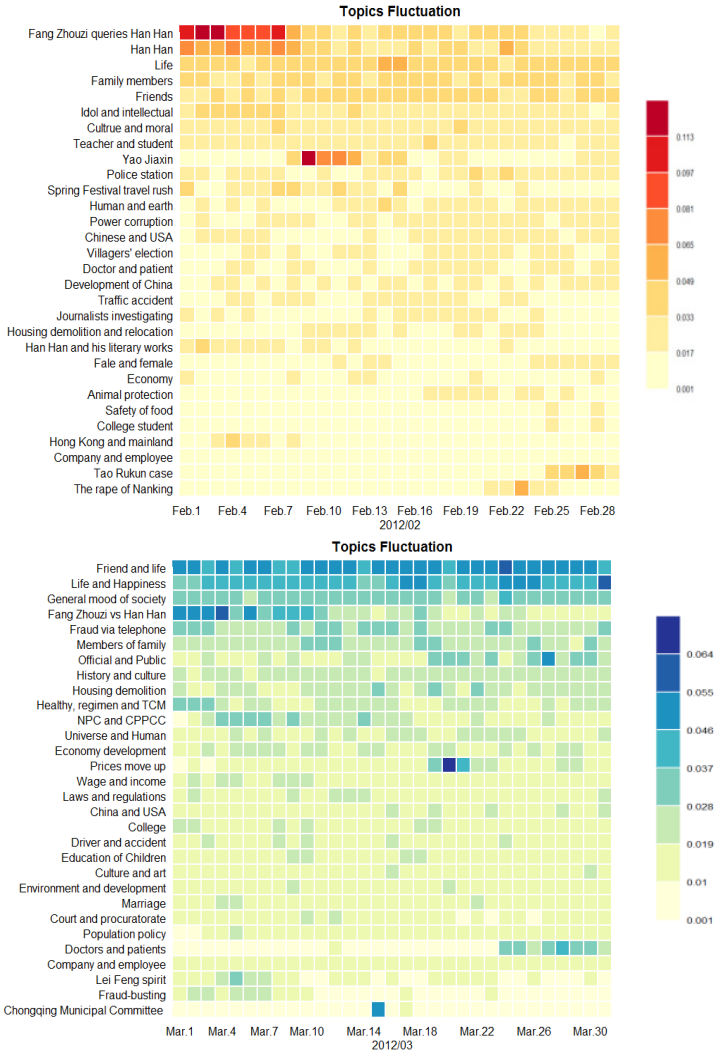


Fig. 2. (*continued*)

In January of 2012, topics about *Spring Festival Gala* became hot due to the lunar new year (January 23). Also, people who care about the tickets ordering during the festival had complaints about the *travel rush* and the *booking website*. Some social events, such as the inappropriate comments given by both Zhang Shaogang and Kong Qingdong, caused public criticisms which last for a short time. The topics on *Han Han*[3] and his debate with *Fang Zhouzi* started, and suddenly became hot in the latter part of the month.

[3] Han Han, a ricing driver and novelist, was in debate with a fraud fighter, Fang Zhouzi, that whether his blogs were written by others.

In February of 2012, topics about *Han Han and Fang Zhouzi* became quite hot, lead that many other related issues were discussed by public. The name *Yao Jiaxin* was mentioned again by the public due to the dispute of civil compensation aroused in February 8. Another event, *Tao Rukun* case, a high school student set fire to a girl who rejected him, caused people in an uproar.

In March of 2012, the topic on *Fang and Han* kept hot in early of this month and cooled down slowly in latter part of the month. The topics about political conferences, NPC (National People's Congress) and CPPCC (Chinese People's Political Consultative Conference) hold during March 3-14 drew the concerns of public. Meanwhile, those topics attracted attentions on the political changes in Chongqing and the relationship between *official and public*. The *rise of oil price* issued by Development and Reform Commission in March 19 led to an eruption of discussion, but it did not last too long. Another example of topic related to time was the discussion on *Lei Feng spirit*.

In this session, results from a series of experiments illustrate online community have the following characteristics: 1) topics about *life, family, friends* and *society morality* are continuous highly concerns by people. Obviously that common people concern their own lives mostly. *Education, festivals, college student, doctor and patient, traffic, economy, culture, official and public* etc. remain as common societal topics which are related to people's life. 2) when the social events happened are related to people's safety or benefits, such as *food safety, travel rush, price rises, school bus,* people discuss real time online; they are quite sensitive to the government decisions and social events contain societal risk.

4 Conclusion

Catching both the central topics and the trend of topics is important for the social management for China. In this paper, we apply dynamic topic models to analyze new posts on forum to discover dynamic topics and their tendency over time. Obviously, DTM offer new ways to browse large and unstructured document collections. Besides the analysis by topic modeling, other methods, such as Self-Organizing Map (SOM) and iView analysis, also can be applied to extract in a meta-synthetic perspective.

Unlike the researches on on-line behavior analysis, such as posting behavior and browsing behavior analysis [14], or hits and replies statistics [15], we concern more on contents and try to find the general regularity and characteristics. The future of our work is that the updated posts which reflect people's discussion on social events will be analyzed to understand the tendency of hot topics better.

Acknowledgements. This research is supported by National Basic Research Program of China under Grant No. 2010CB731405 and National Natural Science Foundation of China under Grant No.71171187.

References

1. Zheng, R., Shi, K., Li, S.: The Influence Factors and Mechanism of Societal Risk Perception. In: Zhou, J. (ed.) Complex 2009, Part II. LNICST, vol. 5, pp. 2266–2275. Springer (2009)
2. Zhang, Z.D., Tang, X.J.: A Preliminary Study of Web Mining for Tianya Forum. In: Proceedings of the 11th Youth Conference of Systems Science and Management Science and 7th Conference of Logistic Systems Technology, pp. 199–204. Wuhan University of Science and Engineering Press, Wuhan (2011) (in Chinese)
3. Tang, X.J., Luo, B.: Understanding College Students' Thought Toward Social Events by Qualitative Meta-Synthesis Technologies. International Journal of Organizational and Collective Intelligence 2(4), 15–30 (2011)
4. Tang, X.J.: Qualitative Meta-synthesis Techniques for Analysis of Public Opinions for in-depth Study. In: Zhou, J. (ed.) Complex 2009, Part II. LNICST, vol. 5, pp. 2338–2353. Springer (2009)
5. Blei, D.M., Lafferty, J.D.: A correlated topic model of Science. J. Annals of Applied Statistics 1, 17–35 (2007)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. J. Journal of Machine Learning Research 3, 993–1022 (2003)
7. Hall, D., Jurafsky, D., Manning, C.D.: Studying the history of ideas using topic models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 363–371. Association for Computational Linguistics (2008)
8. Blei, D.M., Lafferty, J.D.: Dynamic Topic Models. In: Proceedings of the 23rd International Conference on Machine Learning (2006)
9. Alsumait, L., Barbara, D., Domeniconi, C.: On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: Eighth IEEE International Conference on ICDM 2008, pp. 3–12. IEEE (2008)
10. Wang, X., McCallum, A.: Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In: KDD 2006, USA (2006)
11. Wang, C., Blei, D., Heckerman, D.: Continuous time dynamic topic models. In: Uncertainty in Artificial Intelligence, UAI (2008)
12. Meng, C., Zhang, M., Guo, W.: Evolution of Movie Topics Over Time (2012), http://cs229.stanford.edu/projects2012.html
13. Wu, D., Tang, X.J.: Preliminary analysis of Baidu hot words. In: Proceedings of the 11th Youth Conference of Systems Science and Management Science, pp. 478–483. Wuhan University of Science and Engineering Press, Wuhan (2011) (in Chinese)
14. Cui, L.J., He, H., Liu, W.: Research on Hot Issues and Evolutionary Trends in Network Forums. International Journal of u- and e- Service, Science and Technology 6(2), 89–97 (2013)
15. Chen, X., Li, J., Li, S., Wang, Y.: Hierarchical Activeness State Evaluation Model for BBS Network Community. In: 7th International ICST Conference on Communications and Networking in China (CHINACOM), pp. 206–211. IEEE (2012)

Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model

Vivek Narayanan, Ishan Arora, and Arjun Bhatia

Department of Electronics Engineering,
Indian Institute of Technology (BHU), Varanasi, India
{vivek.narayanan.ece09,ishan.arora.ece09,
arjun.bhatia.ece09}@iitbhu.ac.in

Abstract. We have explored different methods of improving the accuracy of a Naive Bayes classifier for sentiment analysis. We observed that a combination of methods like effective negation handling, word n-grams and feature selection by mutual information results in a significant improvement in accuracy. This implies that a highly accurate and fast sentiment classifier can be built using a simple Naive Bayes model that has linear training and testing time complexities. We achieved an accuracy of 88.80% on the popular IMDB movie reviews dataset. The proposed method can be generalized to a number of text categorization problems for improving speed and accuracy.

Keywords: Sentiment classification, Negation Handling, Mutual Information, Feature Selection, n-grams.

1 Introduction

Among the most researched topics of natural language processing is sentiment analysis. Sentiment analysis involves extraction of subjective information from documents like online reviews to determine the polarity with respect to certain objects. It is useful for identifying trends of public opinion in the social media, for the purpose of marketing and consumer research. It has its uses in getting customer feedback about new product launches, political campaigns and even in financial markets [14]. It aims to determine the attitude of a speaker or a writer with respect to some topic or simply the contextual polarity of a document. Early work in this area was done by Turney and Pang ([2], [7]) who applied different methods for detecting the polarity of product and movie reviews.

Sentiment analysis is a complicated problem but experiments have been done using Naive Bayes, maximum entropy classifiers and support vector machines. Pang et al. found the SVM to be the most accurate classifier in [2]. In this paper we present a supervised sentiment classification model based on the Naïve Bayes algorithm.

Naïve Bayes is a very simple probabilistic model that tends to work well on text classifications and usually takes orders of magnitude less time to train when compared to models like support vector machines. We will show in this paper that a high degree of accuracy can be obtained using Naïve Bayes model, which is comparable to the current state of the art models in sentiment classification.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 194–201, 2013.

2 Data

We used a publicly available dataset of movie reviews from the Internet Movie Database (IMDb) [1] which was compiled by Andrew Maas et al. It is a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. Both the training and test sets have an equal number of positive and negative reviews. We chose movie reviews as our data set because it covers a wide range of human emotions and captures most of the adjectives relevant to sentiment classification. Also, most existing research on sentiment classification uses movie review data for benchmarking.

We used the 25,000 documents in the training set to build our supervised learning model. The other 25,000 were used for evaluating the accuracy of our classifier.

3 Naïve Bayes Classifier

A Naive bayes classifier is a simple probabilistic model based on the Bayes rule along with a strong independence assumption.

The Naïve Bayes model involves a simplifying conditional independence assumption. That is given a class (positive or negative), the words are conditionally independent of each other. This assumption does not affect the accuracy in text classification by much but makes really fast classification algorithms applicable for the problem. Rennie et al discuss the performance of Naïve Bayes on text classification tasks in their 2003 paper. [6]

In our case, the maximum likelihood probability of a word belonging to a particular class is given by the expression:

$$P(x_i|\, c) = \frac{Count\ of\ x_i\ in\ documents\ of\ class\ c}{Total\ no\ of\ words\ in\ documents\ of\ class\ c} \tag{1}$$

The frequency counts of the words are stored in hash tables during the training phase.

According to the Bayes Rule, the probability of a particular document belonging to a class c_i is given by,

$$P(c_i|d) = \frac{P(d\ |c_i) * P(c_i)}{P(d)} \tag{2}$$

If we use the simplifying conditional independence assumption, that given a class (positive or negative), the words are conditionally independent of each other. Due to this simplifying assumption the model is termed as "naïve".

$$P(c_i|d) = \frac{(\prod P(x_i|c_j)) * P(c_j)}{P(d)} \tag{3}$$

Here the xi s are the individual words of the document. The classifier outputs the class with the maximum posterior probability.

We also remove duplicate words from the document, they don't add any additional information; this type of naïve bayes algorithm is called Bernoulli Naïve Bayes. Including just the presence of a word instead of the count has been found to improve performance marginally, when there is a large number of training examples.

4 Laplacian Smoothing

If the classifier encounters a word that has not been seen in the training set, the probability of both the classes would become zero and there won't be anything to compare between. This problem can be solved by Laplacian smoothing

$$P(x_i|c_j) = \frac{Count(x_i) + k}{(k+1) * (No\ of\ words\ in\ class\ c_j)} \tag{4}$$

Usually, k is chosen as 1. This way, there is equal probability for the new word to be in either class. Since Bernoulli Naïve Bayes is used, the total number of words in a class is computed differently. For the purpose of this calculation, each document is reduced to a set of unique words with no duplicates.

5 Negation Handling

Negation handling was one of the factors that contributed significantly to the accuracy of our classifier. A major problem faced during the task of sentiment classification is that of handling negations. Since we are using each word as feature, the word "good" in the phrase "not good" will be contributing to positive sentiment rather that negative sentiment as the presence of "not" before it is not taken into account.

To solve this problem we devised a simple algorithm for handling negations using state variables and bootstrapping. We built on the idea of using an alternate representation of negated forms as shown by Das & Chen in [3]. Our algorithm uses a state variable to store the negation state. It transforms a word followed by a not or n't into "not_" + word. Whenever the negation state variable is set, the words read are treated as "not_" + word. The state variable is reset when a punctuation mark is encountered or when there is double negation. The pseudo code of the algorithm is described below:

PSEUDO CODE:.

```
negated := False
for each word in document:
   if negated = True:
       Transform word to "not_" + word.
   if word is "not" or "n't":
       negated := not negated
   if a punctuation mark is encountered
       negated := False.
```

Since the number of negated forms might not be adequate for correct classifications. It is possible that many words with strong sentiment occur only in their normal forms in the training set. But their negated forms would be of strong polarity.

We addressed this problem by adding negated forms to the opposite class along with normal forms of all the features during the training phase. That is to say if we encounter the word “good” in a positive document during the training phase, we increment the count of “good” in the positive class and also increment the count of “not_good” for the negative class. This is to ensure that the number of “not_” forms are sufficient for classification. This modification resulted in a significant improvement in classification accuracy (about 1%) due to bootstrapping of negated forms during training. This form of negation handling can be applied to a variety of text related applications.

6 n-Grams

Generally, information about sentiment is conveyed by adjectives or more specifically by certain combinations of adjectives with other parts of speech.

This information can be captured by adding features like consecutive pairs of words (bigrams), or even triplets of words (trigrams). Words like "very" or "definitely" don't provide much sentiment information on their own, but phrases like "very bad" or "definitely recommended" increase the probability of a document being negatively or positively biased. By including bigrams and trigrams, we were able to capture this information about adjectives and adverbs. Using bigrams and trigrams require a substantial amount of data in the training set, but this is not a problem as our training set had 25,000 reviews. But the data may not be enough to add 4-grams, as this may over-fit the training set. The counts of the n-grams were stored in a hash table along with the counts of unigrams.

7 Feature Selection

Feature selection is the process of removing redundant features, while retaining those features that have high disambiguation capabilities.

The use of higher dimensional features like bigrams and trigrams presents a problem, that of the number of features increasing from 300,000 to about 11,000,000. Most of these features are redundant and noisy in nature. Including them would affect both efficiency and accuracy. A basic filtering step of removing the features/terms which occur only once is performed. Now the number of features is reduced to about 1,500,000 features. The features are then further filtered on the basis of mutual information [3].

7.1 Mutual Information

Mutual information is a quantity that measures the mutual dependence of the two random variables. Formally, the mutual information of two discrete random variables X and Y can be defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right), \tag{5}$$

Where p(x,y) is the joint probability distribution function of X and Y, and P(X) and P(Y) are the marginal probability distribution functions of X and Y respectively. Here X is an individual feature which can take two values, the feature is present or absent and Y is the class, positive or negative. We selected the top k features with maximum mutual information. By plotting a graph between accuracy and number of features, the optimal value for k was found out to be 32,000.

A plot of **Accuracy versus Number of features** is shown in Fig 1:

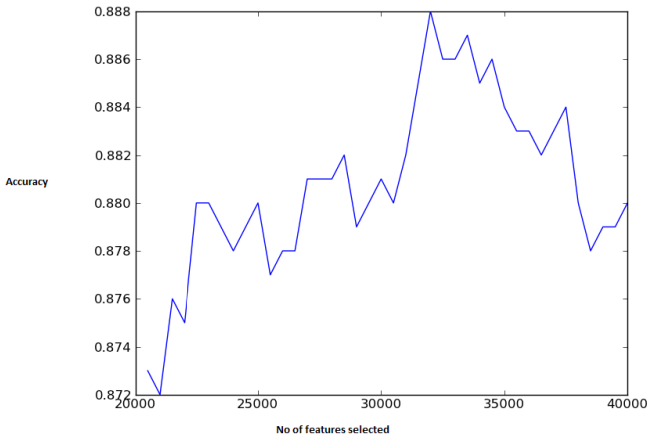


Fig. 1. Plot of Accuracy v/s No of features selected on Validation set

8 Results

We implemented the classifier in Python using hash tables to store the counts of words in their respective classes. The code is available at [13]. Training involved preprocessing data and applying negation handling before counting the words. Since we were using Bernoulli Naive Bayes, each word is counted only once per document. On a laptop running an Intel Core 2 Duo processor at 2.1 GHz, training took around 1 minute 30 seconds and used about 700 megabytes of memory. The memory usage stems largely from bigrams and trigrams prior to feature selection.

The optimal number of features was chosen by using a validation set of 1000 documents, the plot of accuracy v/s number of features is shown in Fig 1. Then the accuracy was measured on the entire test set of 25000 documents. The time for feature selection was about 3 minutes.

8.1 Performance and Comparison

We obtained an overall classification accuracy of 88.80% on the test set of 25000 movie reviews. The running time of our algorithm is O(n + V lg V) for training and

O(n) for testing, where n is the number of words in the documents (linear) and V the size of the reduced vocabulary. It is much faster than other machine learning algorithms like Maxent classification or Support Vector Machines which take a long time to converge to the optimal set of weights. The accuracy is comparable to that of the current state-of-the-art algorithms used for sentiment classification on movie reviews. It achieved a better or similar accuracy when compared to more complicated models like SVMs, autoencoders, contextual valence shifters, matrix factorisation, appraisal groups etc used in [2], [7], [8], [9], [10], [11] on the dataset of IMDb movie reviews.

8.2 Results Timeline

The table and graph illustrate the evolution of the accuracy of our classifier and how the inclusion of certain features helped.

Table 1. Results Timeline

Feature Added	Accuracy on test set
Original Naive Bayes algorithm with Laplacian Smoothing	73.77%
Handling negations	82.80%
Bernoulli Naive Bayes	83.66%
Bigrams and trigrams	85.20%
Feature Selection	88.80%

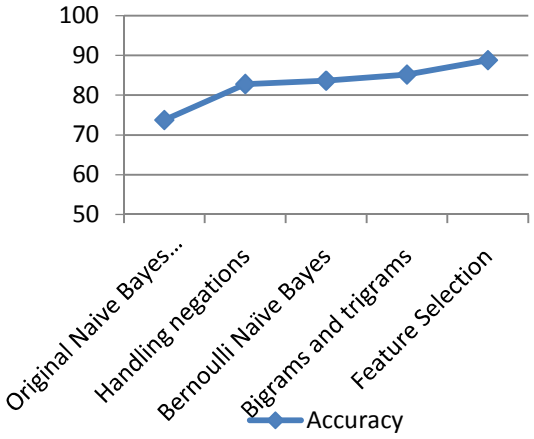


Fig. 2. Evolution of classification accuracy

9 Conclusion

Our results show that a simple Naive Bayes classifier can be enhanced to match the classification accuracy of more complicated models for sentiment analysis by choosing the right type of features and removing noise by appropriate feature selection. Naive Bayes classifiers due to their conditional independence assumptions are extremely fast to train and can scale over large datasets. They are also robust to noise and less prone to overfitting. Ease of implementation is also a major advantage of Naive Bayes. They were thought to be less accurate than their more sophisticated counterparts like support vector machines and logistic regression but we have shown through this paper that a significantly high accuracy can be achieved. The ideas used in this paper can also be applied to the more general domain of text classification.

Acknowledgement. We would like to express our sincere gratitude to our mentor, Professor R. R. Das for sparing his valuable time and guiding us during the project. We are also thankful to Prof. P. K. Mukherjee for his support and encouragement. Lastly, we would like to thank all the faculty members and support staff of the Department of Electronics Engineering, IIT (BHU).

References

1. Large Movie Review Dataset (n.d.), `http://ai.stanford.edu/~amaas/data/sentiment/`
2. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10. Association for Computational Linguistics (2002)
3. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
4. Das, S., Chen, M.: Yahoo! for Amazon: Sentiment parsing from small talk on the web. In: EFA 2001 Barcelona Meetings (2001)
5. Pauls, A., Klein, D.: Faster and smaller n-gram language models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (2011)
6. Rennie, J.D., et al.: Tackling the poor assumptions of naive bayes text classifiers. In: Machine Learning-International Workshop then Conference, vol. 20(2) (2003)
7. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning Word Vectors for Sentiment Analysis. In: The 49th Annual Meeting of the Association for Computational Linguistics, ACL 2011 (2011)
8. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Computational Intelligence 22(2), 110–125 (2006)
9. Li, T., Zhang, Y., Sindhwani, V.: A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, vol. 1. Association for Computational Linguistics (2009)

10. Matsumoto, S., Takamura, H., Okumura, M.: Sentiment classification using word subsequences and dependency sub-trees. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 301–311. Springer, Heidelberg (2005)
11. Springer, Heidelberg
12. 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. ACM (2005)
13. Socher, R., et al.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2011)
14. Source code of classifier developed for this paper, http://github.com/vivekn/sentiment
15. Devitt, A., Ahmad, K.: Sentiment polarity identification in financial news: A cohesion-based approach. In: Annual Meeting-Association for Computational Linguistics, vol. 45(1) (2007)
16. Peng, F., Schuurmans, D.: Combining naive Bayes and n-gram language models for text classification. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 335–350. Springer, Heidelberg (2003)

A Scale-Free Based Memetic Algorithm for Resource-Constrained Project Scheduling Problems

Lixia Wang and Jing Liu

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China
kjdxwlx@126.com, neouma@mail.xidian.edu.cn

Abstract. The resource-constrained project scheduling problem (RCPSP) is a popular problem that has attracted attentions of many researchers with various backgrounds. In this paper, a new memetic algorithm (MA) based on scale-free networks is proposed for solving RCPSPs, namely SFMA-RCPSPs. In SFMA, the chromosomes are located on a scale-free network. Thus, each chromosome can only communicate with the ones that have connections with it. In the experiments, benchmark problems, namely Patterson, J30 and J60, are used to validate the performance of SFMA. The results show that the SFMA performs well in finding out the best known solutions especially for Patterson and J30 data sets, besides, the average deviations from the best known solutions are small. Therefore, SFMA improves the search speed and effect.

Keywords: scale free networks, memetic algorithms, local search, resource-constrained project scheduling.

1 Introduction

Usually, a project is made up of several tasks and each task needs time and resources to complete. When the gross of resource is limited, and there occurs precedence relationships among these tasks, it comes the resource-constrained project scheduling problem (RCPSP). We can image this problem as a generalization of the well-known job shop scheduling problem, and it belongs to the class of combinatorial problems that is most difficult to solve, i.e., it is NP-hard [1].

In recent years, the research on scale-free networks becomes more and more popular [2]. In a scale-free network, most vertices have few connections with other nodes but there exist a few of vertices that have large amounts of connections [3], then these vertices with relatively large connections play an important part in communication of information in the network [2]. Various of networks in real world have the scale-free properties, such as World Wide Web [4], neural networks [5], metabolic interaction networks [6]. Therefore, more and more researchers propose to combine scale-free networks with existing algorithms to solve optimization problems [2].

Evolutionary algorithms (EAs), a kind of stochastic global optimization methods inspired by the biological mechanism of evolution and heredity, have been successfully used to solve various problems [7-12]. Memetic algorithms (MAs) are proposed by

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 202–209, 2013.

Pablo Moscato [13], and are also called genetic local search, cultural algorithms or hybrid genetic algorithms [14]. In fact, MAs are the combination of population-based global search and the heuristic local search [15-17]. This kind of combination makes the search efficiency of MAs outperform EAs. Therefore, we propose using MAs to solve RCPSPs. The SFMA is tested on the Patterson, J30 and J60 sets, the results obtained show that it performs well in improving the search efficiency.

Many methods have been proposed for RCPSPs, such as heuristics methods, they can be divided into priority rule-based X-pass methods, non-standard metaheuristics and classical metaheuristics etc [18], Classical metaheuristics consist of genetic algorithm, simulated annealing algorithm, tabu search algorithm, particle swarm optimization algorithm, ant colony optimization etc [19]. Alcaraz and Maroto propose a GA based on the serial SGS and list representation [20]. Valls et al. develop a SA method focuses on forward-backward improvement [21]. Fleszar and Hindi use a variable neighborhood search to solve the RCPSP [1].

The rest of this paper is organized as follows. Section 2 introduces the detail content about the SFMA. Section 3 gives the experiments on the benchmark data sets. Section 4 concludes the work in this paper.

2 Scale-Free Based Memetic Algorithm for RCPSP

2.1 RCPSPs

In a resource-constrained project scheduling problem, a project is made up of n tasks which are marked by 1, 2, …, n. A duration d_j is needed to finish task j [22]. The objective of RCPSPs is to arrange the start time for these tasks. In the same time, two kinds of constraints exist to bound the work; they are called precedence constraints which can be depicted by a graph shown in Figure 1 and resource capacities. The precedence constraints mean that a task can not be arranged before all of its predecessors have been finished [23]. Every process to complete a task needs certain of time and resources. The aim is to find out start and finish time for all of the tasks, at the same time, both the constraints given above must be satisfied and the duration of the project or the resources cost or some other suitable objective is optimized [5]. Makespan is the most popular objective function for RCPSP, because people would like to use the least time to complete a project. In this paper, makespan is used as the objective function.

As shown in figure 1, each vertex represents a task, task 1 and n are dummy tasks with their durations d_1 and d_n both be zeros. Task 1 and n are called the initial task and final task which must be arranged before and after all of the other tasks respectively. If there is an arrow between task i and j of length d_i, it presents that task i is j's predecessor and j can not start before i has finished, d_i is the duration of i. All immediate successors of the task i make up of a set represented by A_i, in the same way, the set of task i's immediate predecessors is denoted by B_i [24]. We use γ_i to presents the amount of predecessors of task i, it equals to the number of arrows terminate at vertex i in figure 1. The graph thus defined must be acyclic; otherwise, someone will be its own predecessor [7].

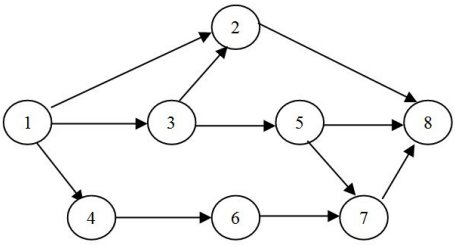


Fig. 1. Precedence graph

2.2 Population Initializations with Scale-Free Structure

For a node k in a network, its degree distribution usually expressed by $P(k)$. In traditional random networks just like ER networks, the degree distribution obeys the Poisson distribution [3]. In fact, the distribution of real networks is far from the Poisson distribution [25], it is similar to a simple form:

$$P(k) \sim k^{-r} \tag{1}$$

Here γ is a constant exponent which varies between 1.8 and 2.5 [3]. Its speed of falling off is much slower than ER networks, so it allows a few nodes to have large connections. Researchers call this distribution a power-law distribution [3].

Before initializing the population, firstly construct the scale free network to deposit the chromosomes. The network is constructed according to BA-modal, which is proposed by Barabási and Albert [25].

Algorithm 1. Construction of the BA modal

Step 1: Produce a network with m_0 nodes attach to each other;

Step 2: When a new node is generated, calculate the degree distribution of these existing nodes, then select which m nodes (node) to attach according to the distribution, here m and m_0 are both integers and $m<m_0$;

Step 3: Execute step 2 repeatedly, until the total number of nodes is enough.

After finishing the construction of scale free network, the chromosomes are generated corresponding to the vertices on the network. Three methods are used to produce the initial population alternately until every vertex on the network has a chromosome and the corresponding. The first two methods are recomposed from the famous algorithms [7], and the third one is proposed by us, here we just give the detail procedure of the third one.

Suppose all the tasks are presented in a nonperiodic graph G, each task corresponds to one node, if there is an arrow from node i to node j ($i, j \in G$), thus task i is called the immediate predecessor of task j, or task j is the immediate successor of task i. γ_i is used to present the number of immediate successors of task i.

Algorithm 2. Method 3 for initialization

Step 1: For each task v of G, calculate its number of immediate predecessors γ_w, s is an empty set use to deposit tasks;

Step 2: Find out the tasks those γ_s are zeros then select only one task w randomly and arrange it into s.

Step 3: For the task arranged just now, find out its immediate successors for example task v, then $\gamma_v = \gamma_v - 1$;

Step 4: Find out the tasks that γ=0 and belong to the immediate successors of the task arranged just now, then arrange one of them randomly, repeat step 3 and 4 until no task can be arranged;

Step 5: Repeat step 2, 3 and 4 until all of the tasks in G have been arranged into s, then a chromosome has been produced.

2.3 Competition-Crossover Operator

As all of the chromosomes are fixed on a scale-free network, it represents that a chromosome can change information only with the one who has connection relationship with it. For each chromosome s in the population, we firstly find out the units connect with it and compare those fitness values, the unit has the smallest makespan is marked as m, let m execute the mutation operator then obtain a new chromosome labeled p_2, the mutation operator will be introduced in the content below, let s be p_1. At present if a random number produced is larger than the crossover probability Pc, we select the makespan smaller one between p_1 and p_2 as a new unit and mark by s_{new}, then s will be replaced by s_{new}; otherwise, if the random number is smaller than P_c, the crossover operator will be executed, here we use the two-point operator adapted from [7], the detail procedure is introduced below.

Algorithm 3. Crossover Operator

Let p_1 and p_2 be the two parents, c_1 and c_2 be the children. n is the total number of the tasks as well as the length of a chromosome.

Step 1: Generate two numbers r_1 and r_2 randomly, $r_1 < r_2$, r_1, r_2 are both integers and belong to interval $(1, n)$.

Step 2: $c_1(1: r_1) = p_1(1: r_1)$ and $c_1(r_2+1: n) = p_1(r_2+1: n)$.

Step 3: for j=1: n, if $p_2(j)$ is not belong to c_1, then $r_1 = r_1+1$ and $c_1(r_1) = p_2(j)$.

Similarly, c_2 is obtained in the same way by interchanging p_1 and p_2. Now there are four units include p_1, p_2, c_1, c_2, then choose the one with the minimum makespan to replace s. In addition, if the parents are valid, the children produced are valid too, because the children inherit the prior orders from their parents.

2.4 Mutation Operator and Local Search Operator

The mutation operator is adapted from [7]. This kind of mutation is helpful to the diversity of the population. For any task in a chromosome, our method to mutate is to find the interval that allows the task to move and will not violate the precedence constraints. The detail of the operator is described as follow; image s is a valid chromosome.

Algorithm 4. Mutation operator

Step 1: Choose a task i among all of the tasks in random except 1 and n, here n indicates the length of the chromosome, then find out its position p_i in the chromosome s;

Step 2: For the tasks belong to i's immediate predecessors, calculate their largest position in the chromosome s marked l; similarly, find out the smallest position of the tasks of i's immediate successors and label as e;

Step 3: Randomly generate an integer k between l and e, and k can not equal to p_i;
Step 4: Then task i can move to position k, and a new chromosome is generated.

In order to keep the better units in current generations, we only receive the mutation ones whose makespans are smaller than before. Finally, compare the fitness value of the new chromosome with the old one's, and reserve the individual with the smaller makespan.

Calculate the fitness values of the current population and find out the minimum makespan, as the number of chromosomes whose fitness values are equal to the minimum makespan is not only one, the SFMA find out the whole chromosomes and execute local search upon them to make their fitness values less and less. Here we let the best individuals in the current generations execute mutation operator 20 times. During the experiments we can see that this operator is much helpful to obtain the optimal makespan rapidly.

2.5 Implementation of SFMA

The overall scheme of the SFMA is demonstrated in Algorithm 7 below. As states above, MAs are GAs combine with local search, competition-crossover and mutation are belonging to the GAs, step 10-12 are local search. The algorithm is stopped when either the minimum makespan is equal to the best-known makespan or the total schedules produced exceed the number we set.

Algorithm 5. The implementation of SFMA
1. Algorithm begin;
2. Construct the scale free network by BA modal;
3. Initialize original population;
4. Calculate the fitness values of the population;
5. Store the minimum makespan of the current population as fmin;
6. while the stopping conditions are not satisfied;
7. Execute competition-crossover operators;
8. Execute mutation operator;
9. Find out the minimum makespan in the current population f_c;
10. for the chromosomes whose fitness values are equal to f_c;
11. Execute local search;
12. end
13. Update the minimum makespan f_{min};
14. end
15. Algorithm end

3 Experiments

In this section, the benchmark problems from PSPLIB [26], namely Patterson, J30, and J60, are used to validate the performance of SFMA. In the following experiments, for each instance, 10 independent runs are executed and the average values of the following criteria are used to show the performance of the SFMA:

(1) %NBS: It means the percentage ratio of the best known solutions found by the SFMA to the total number of instances [7].

(2) %ERR: Means the average error; we calculate this error according the following expression:

$$\%\mathrm{ERR} = \frac{solution\_makespan - best\_makespan}{best\_makespan} \bullet 100\% \tag{6}$$

In the scale free network we generate, *m* indicates the number of vertices (vertex) connected by the new node, we randomly choose one hundred instances from the J30 sets and test the effect of m on %NBS, there we set *m* to be 1, 2, 3, 4, 5, 8 respectively, and other parameters are the same, the results are shown in Table 1, from the results we can see, the smaller m is set, the better it performs. Besides, through the tests on the same data, we find that the two point crossover operator performs better than the one point crossover operator.

Table 1. Comparison of different *m*

m	**1**	**2**	**3**	**4**	**5**	**8**
%NBS	76.0	74.7	73.7	74.3	73.1	74.2

In the local search process, as the number of individuals whose fitness values are equal to the minimum one is unconcern, we choose several individuals to execute local search operator rather than only one individual. The results shown in Table 2 proves this. Through the results, we can see that no matter the %NBS or the %ERR, the multi units method performs much better than the single unit method. Table 3 shows the average deviations, and the stopping criterias are maximum of 1000, 5000 schedules respectively. The results validate that the deviations are not small enough, so in the future, it is a goal for us to reduce the deviation.

Table 2. Comparison of local search with single unit and multi units

Data sets	**Local search**	**%NBS**	**%ERR**	**AvgEvals**
Patterson	Single unit	93.36	0.08	954
Patterson	Multi units	98.73	0.02	735
J30	Single unit	82.68	0.47	1607
J30	Multi units	88.27	0.28	1867
J60	Single unit	67.42	1.78	4130
J60	Multi units	70.04	1.42	4707

Table 3. Deviation from critical-path lower bound

Data sets	**Max schedules**	
	1000	5000
J30	0.94	0.42
J60	14.58	13.06

4 Conclusions

In this paper, we proposed a SFMA for the single project, single mode, resource-constrained project scheduling problem. The SFMA combines the global search and the local search, so that it can search the solution space much more thoroughly. Besides, it is characterized by introducing a scale-free network, here we construct a scale-free network to let the chromosomes to fix on, when executing the competition and crossover operators, the chromosomes can not operate with random individuals but must with the ones attach to them. Thus the good individuals can influence others more. The SFMA is tested on the Patterson, J30 and J60 sets, the average rates of best known solutions found by SFMA are rather high especially on Patterson data sets and J30 sets, it proves that the SFMA improves the search efficiency.

Acknowledgement. This work is partially supported by the National Natural Science Foundation of China under Grants 61271301 and 61103119, and the Fundamental Research Funds for the Central Universities under Grant K5051202052.

References

1. Krzysztof, F., Khalil, S.H.: Solving the resource-constrained project scheduling problem by a variable neighbourhood search. European Journal of Operational Research 155(1), 402–413 (2004)
2. Cheng, Z., Yi, Z.: Scale-free fully informed particle swarm optimization algorithm. Information Sciences 181(20), 4550–4568 (2011)
3. Oliver, H., Michael, S., Wolfgang, K.: Scale-Free Networks the Impact of Fat Tailed Degree Distribution on Diffusion and Communication Processes. Wirtschaftsinformatic, 267–275 (2006)
4. Barabási, A.L., Albert, R., Jeong, H.: Scale-free characteristics of random networks: the topology of the world wide web. Physica A 281(1-4), 69–77 (2000)
5. Egúluz, V.M., Chialvo, D.R., Cencchi, G.A., Baliki, M., Apkarian, A.V.: Scale-free brain functional networks. Phys. Rev. Lett. 94(1) (2005)
6. Jeong, H., Mason, S., Barabási, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. Nature 411, 41–42 (2001)
7. Khalil, S.H., Hongbo, Y., Krazysztof, F.: An Evolutionary Algorithm for Resource-Constrained Project Scheduling. IEEE Transactions on Evolutionary Computation 6(5), 512–518 (2002)
8. Zhong, W., Liu, J., Xue, M., Jiao, L.: A multiagent genetic algorithm for global numerical optimization. IEEE Trans. on Syst., Man, and Cybern., Part B 34(2), 1128–1141 (2004)
9. Liu, J., Zhong, W., Jiao, L.: A multiagent evolutionary algorithm for constraint satisfaction problems. IEEE Trans. on Syst., Man, and Cybern., Part B 36(1), 54–73 (2006)
10. Liu, J., Zhong, W., Jiao, L.: Moving block sequence and organizational evolutionary algorithm for general floorplanning with arbitrarily shaped rectilinear blocks. IEEE Trans. on Evolutionary Computation 12(5), 630–646 (2008)
11. Jiao, L., Liu, J., Zhong, W.: An organizational coevolutionary algorithm for classification. IEEE Trans. on Evolutionary Computation 10(1), 67–80 (2006)

12. Liu, J., Zhong, W., Jiao, L.: A multiagent evolutionary algorithm for combinatorial optimization problems. IEEE Trans. on Systems, Man, and Cybernetics Part B 40(1), 229–240 (2010)
13. Pablo, M.: On evolution, search, optimization, genetic algorithms and martial arts towards memetic algorithms Caltech concurrent computation program. C3P Report (1989)
14. Joshua, D. K., David, W.C.: M-PAES: A memetic algorithm for multiobjective optimization. Evolutionary Computation 1(1), 325–332 (2000)
15. Ong, Y.S., Keane, A.J.: Met-lamarckian learning in memetic algorithms. IEEE Transaction on Evolutionary Computation 8(2), 99–110 (2004)
16. Ong, Y.S., Lim, M.H., Zhu, N.: Classification of adaptive memetic algorithms: a comparative study. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(1), 141–152 (2006)
17. Ong, Y.S., Lim, M., Chen, X.: Memetic computation-Past, present & future (Research Frontier). IEEE Computational Intelligence Magazine 5(2), 24–31 (2010)
18. Rainer, K., Sonke, H.: Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research 174(1), 23–37 (2006)
19. Tormos, P., Lova, A.: An efficient multi-pass heuristic for project scheduling with constrained resources. International Journal of Production Research 41, 1071–1086 (2003)
20. Alcaraz, J., Maroto, C.: A robust genetic algorithm for resource allocation in project scheduling. Annals of Operations Research 102, 83–109 (2001)
21. Valls, V., Ballestin, F., Quintanilla, M.S.: Justification and RCPSP: A technique that pays. European Journal of Operational Research 165, 375–386 (2005)
22. Blazewicz, J., Lenstra, J.K., Rinnooy, A.H.G.: Scheduling subject to resource constraints: Classification and complexity. Discrete Appl. Maths. 5, 11–24 (1983)
23. Valls, V., Ballestin, F., Quintanilla, M.S.: A hybrid genetic algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research 185(2), 495–508 (2008)
24. Tormos, P., Lova, A.: A Competitive Heuristic Solution Technique for Resource-Constrained Project Scheduling. Annals of Operations Research 102, 65–81 (2001)
25. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
26. Kolisch, R., Sprecher, A., Drexl, A.: Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41(10), 1693–1703 (1995)

A Direction based Multi-Objective Agent Genetic Algorithm

Chen Zhu and Jing Liu

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China
zhuchen.038@163.com, neouma@mail.xidian.edu.cn

Abstract. A direction based multi-objective agent genetic algorithm (DMOAGA) is proposed in this paper. In order to take advantage of the effective direction information and depth of local search to mine non-dominated solutions, the direction perturbation operator is also employed. The neighborhood non-dominated solutions are generated using tournament selection and "*average distance*" rule, which maintains the diversity of non-dominated solution set. In the experiments, the benchmark problems UF1~UF6 and ZDT1~ZDT4 are used to validate the performance of DMOAGA. We compared it with NSGA-II and DMEA in terms of generational distance (GD) and inverted generational distance (IGD). The results show that DMOAGA has a good diversity and convergence, the performances on most of benchmark problems are better than DMEA and NSGA-II.

Keywords: Multi-objective optimization problems, Direction information, Multi-agent systems, Genetic algorithms.

1 Introduction

Most of the engineering problems to be optimized have multiple objectives with conflicting among each other. This kind of problems is known as multi-objective optimization problems (MOPs). Each solution represents a compromise among the conflicting objectives. These solutions are Pareto optimal solutions, and the collection of these solutions is called Pareto optimal solution set (POS). Using evolutionary algorithms (EAs) [1-2] to deal with such problems called multi-objective evolutionary algorithms (MOEAs).

Various algorithms have been proposed to solve MOPs with using archives. Yet most of them did not realize that the archive itself has a wealth of knowledge. Therefore, inspired by the work in [3], we make use of direction information in archives to search for the Pareto front (PF). In addition, the multi-agent technology is widely used in distributed computing and optimization problems [4-5]. For function optimization problems, Zhong *et al*. combined genetic algorithms (GAs) with the perception and reaction ability of agents, and proposed a multi-agent genetic algorithm (MAGA) [4] for single-objective optimization problems which is reported to outperform traditional GAs. MAGA′s success shows the potential of the combination of multi-agent systems

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 210–217, 2013.

and genetic algorithms. Therefore, we extend it to multi-objective optimization problems, and propose a direction based multi-objective agent genetic algorithm (DMOAGA).

There are 5 operators in DMOAGA, where competition, mutation and crossover operators ensure the search efficiency of multi-objectives algorithm, and direction perturbation, self-learning operators help to find as many as Pareto solutions which are able to approximate PF. Each operator in the algorithm makes very important contribution to the results. The key step in the multi-agent systems for solving multi-objective problems is to find neighborhood non-dominated solutions. These non-dominated solutions constitute a global elite archive, and the rich direction information of archive also be used in the evolutionary process.

2 Multi-Agent Genetic Algorithm

According to [6-7], an agent is a physical or virtual entity which is able to live and act in the environment and sense its local environment. Multi-agent systems are computational systems in which several agents interact together in order to achieve certain purposes. The environment is organized as a lattice-like structure that all agents living is defined as follows:

Definition 1. All agents live in a lattice-like environment, $\boldsymbol{L}$, which is called an agent lattice. The size of $\boldsymbol{L}$ is $\boldsymbol{L}_{size}\times\boldsymbol{L}_{size}$, where $\boldsymbol{L}_{size}$ is an integer. Each agent is fixed on a lattice-point and it can only interact with its neighbors. Suppose that the agent located at (i,j) is represented as $\boldsymbol{L}_{i,j}$, $i, j=(1, 2, \ldots, \boldsymbol{L}_{size})$, then the neighbors of $\boldsymbol{L}_{i,j}$, $\mathbf{Neighbor}_{i,j}$, are defined as follow:

$$\mathbf{Neighbor}_{i,j}=\{\boldsymbol{L}_{i',j'}, \boldsymbol{L}_{i',j}, \boldsymbol{L}_{i',j''}, \boldsymbol{L}_{i,j'}, \boldsymbol{L}_{i,j''}, \boldsymbol{L}_{i'',j'}, \boldsymbol{L}_{i'',j}, \boldsymbol{L}_{i'',j''}\} \tag{1}$$

As shown in Fig.1: there are 8 agents form a neighborhood in agent system, where

$$i'=\begin{cases} i-1, i\neq 1 \\ L_{size}, i=1 \end{cases}, \quad j'=\begin{cases} j-1, j\neq 1 \\ L_{size}, j=1 \end{cases}, \quad i''=\begin{cases} i+1, i\neq L_{size} \\ 1, i=L_{size} \end{cases}, \quad j''=\begin{cases} j+1, j\neq L_{size} \\ 1, j=L_{size} \end{cases}.$$

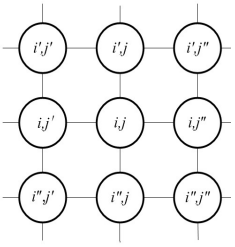


Fig. 1. Model of agent neighborhood

Perception of the local environment of each agent will make it exchange information with its neighbors [8]. More detailed definitions of agent are available in [4].

A new definition about whether an agent survival in the multi-objective optimization problems is follows:

Definition 2. Assume that $\boldsymbol{L}_{i,j}=(l_1, l_2, \ldots, l_n)$ is an agent living in lattice, and $\mathbf{Max}_{i,j}=(m_1, m_2, \ldots, m_n)$ is a Pareto solution among the neighbors of $\boldsymbol{L}_{i,j}$, $\mathbf{Max}_{i,j}\in\mathbf{Neighbor}_{i,j}$, thus $\forall\boldsymbol{a}\in\mathbf{Neighbor}_{i,j}$, $\mathbf{Energy}_r(\boldsymbol{a})\leq\mathbf{Energy}_r(\mathbf{Max}_{i,j})$ and there exists at least one objective makes $\mathbf{Energy}_t(\boldsymbol{a})<\mathbf{Energy}_t(\mathbf{Max}_{i,j})$, $r, t=(1, 2, \ldots, k)$, where $\mathbf{Energy}_k(\boldsymbol{a})$ is the k-th objective of agent $\boldsymbol{a}$. Mathematically, that is marked as $\boldsymbol{a}\prec\mathbf{Max}_{i,j}$, means $\boldsymbol{a}$ dominate $\mathbf{Max}_{i,j}$.

In DMOAGA, neighborhood competition and neighborhood orthogonal crossover operator can be seen as competition and cooperation between agents; while mutation operator can be seen as the ability that an agent uses its knowledge to survive.

Neighborhood competition operator: If $\boldsymbol{L}_{i,j} \prec \mathbf{Max}_{i,j}$, $\boldsymbol{L}_{i,j}$ survives in lattice; otherwise $\boldsymbol{L}_{i,j}$ is the loser and must die. Then empty lattice point will be occupied by $\mathbf{Max}_{i,j}$. $\mathbf{Max}_{i,j}$ has two strategies defined in MAGA [4] to occupy the lattice and it selects them with probability P_o. If $\boldsymbol{U}(0,1)<P_o$, occupying strategy **1** is selected; otherwise, occupying strategy **2** is selected, where, $\boldsymbol{U}(0,1)$is a random number of uniformly distributed in interval [0,1].

Then generated two values i_1 and i_2 randomly and satisfy $1<i_1<n$, $1<i_2<n$, $i_1<i_2$, such that $\mathbf{New}'_{i,j}=(e'_1, e'_2, \ldots, e'_n)$ is determined by (2)

$$\mathrm{New}'_{i,j} = (m'_1, m'_2, \ldots, m'_{i_1-1} m'_{i_2}, m'_{i_2-1}, \ldots, m'_{i_1+1}, m'_{i_1}, m'_{i_2+1}, m'_{i_2+2}, \ldots, m'_n) \tag{2}$$

Finally, $\mathbf{New}_{i,j}=(e_1, e_2, \ldots, e_n)$ is obtained by mapping $\mathbf{New}'_{i,j}$ back to interval $[\underline{x}_k, \overline{x}_k]$ according to

$$e_k=\underline{x}_k+e'_k\cdot(\overline{x}_k-\underline{x}_k),\ k=(1, 2, \ldots, n) \tag{3}$$

The twice transform combine random search with information of victor skillfully and correspond to the exploration operator of genetic algorithms.

Neighborhood Orthogonal Crossover Operator: The orthogonal crossover operator generates new individuals by means of the orthogonal design [9]. In DMOAGA, it is performed on $\boldsymbol{L}_{i,j}$ and $\mathbf{Max}_{i,j}$ to achieve the purpose of cooperation. The search space division is same as MAGA.

Then (F-1) integers, $1<k_1<k_2<\ldots<k_{F-1}<n$, are generated randomly. The F factors for each agent $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)$ are created as follows

$$f_1(a_1,\ldots,a_{k_1}),\ f_2(a_{k_1+1},\ldots,a_{k_2}),\ \ldots,\ f_F(a_{k_{F-1}+1},\ldots,a_{k_n}) \tag{4}$$

$\boldsymbol{Q}_2$ levels for the i-th factor f_i are defined as follows:

$$f_i(1)=(\beta_{k_{i-1}+1,1},\beta_{k_{i-1}+2,1},\ldots,\beta_{k_i,1}),\ f_i(2)=(\beta_{k_{i-1}+1,2},\beta_{k_{i-1}+2,2},\ldots,\beta_{k_i,2}),\ \ldots,\ f_i(Q_2)=(\beta_{k_{i-1}+1,Q_2},\beta_{k_{i-1}+2,Q_2},\ldots,\beta_{k_i,Q_2}) \tag{5}$$

Finally, the orthogonal array $L_{M_2}=(Q_2^F)=[b_{i,j}]_{M_2\times F}$ is applied to generate $\boldsymbol{M}_2$ agents.

$$(f_1(b_{1,1}),f_2(b_{1,2}),\ldots,f_F(b_{1,F})),\ (f_1(b_{2,1}),f_2(b_{2,2}),\ldots,f_F(b_{2,F})),\ \cdots,\ (f_1(b_{M_2,1}),f_2(b_{M_2,2}),\ldots,f_F(b_{M_2,F})) \tag{6}$$

Then the neighbor non-dominated agent among the $\boldsymbol{M}_2$ agents is selected to replace $\boldsymbol{L}_{i,j}$. Here, $\boldsymbol{Q}_2$, $\boldsymbol{F}$ and $\boldsymbol{M}_2$ are fixed at 3, 4, 9, respectively. The orthogonal array is $\boldsymbol{L}_9(3^4)$ is same as using in [9].

Mutation Operator: A new agent, $\mathbf{New}_{i,j}=(e_1, e_2, \ldots, e_n)$, is generated as

$$e_k=\begin{cases} l_k, & U(0,1)<1/n \\ l_k+G(0,1/t), & \text{otherwise} \end{cases},\ k=(1, 2, \ldots, n) \tag{7}$$

where, $\boldsymbol{G}(0,1/t)$ is Gaussian random number generator, t is evolution generation. Then, substitute $\boldsymbol{L}_{i,j}$ with $\mathbf{New}_{i,j}$.

3 Direction Based Multi-Objective Agent Genetic Algorithm

3.1 Direction Perturbation Operator

In [16], an external archive has been created to save non-dominated solutions that generated in the process of evolution and used to produce next generation solution together with the newly generated offspring. In DMEA, it makes use of deeper archival information, so it guides the evolutionary process by extracting direction information that can improve solutions. The archive in DMOAGA not only contributes solutions to the next generation, but also supports the derivation of directions for offspring production. Offspring are generated in a process in which randomly-selected parents are perturbed along identified directions of improvement. Two types of perturbation direction as [3] are used.

Convergence Direction (CD): Given a dominated parent Par_d with decision space coordinates $Par_d(i)$, i=(1, 2, …, n), a perturbation rate $0<P_e<1$, and a normalized CD d_1, we perturb Par_d to form a vector S_1 as follows:

$$S_1(i) \equiv Par_d(i) + RND_i(p)\sigma_1 d_1(i),\ \ d_1 \equiv A_1 - Par_d \big/ |A_1 - Par_d|,\ \ d_1 \equiv A_1 - Par_d \big/ |A_1 - Par_d| \tag{8}$$

where σ_1 is a scalar model parameter which is uniformly sampled from interval (0,2) at random. If $U(0,1)<P_e$, $RND_i(p)$ equals 1; otherwise 0.

Spread Direction (SD): Given a non-dominated parent Par_n with decision space coordinates $Par_n(i)$, i=(1, 2, …, n), three solutions A_1, A_2, A_3 randomly selected from the archive and a normalized SD d_2, we perturb Par_n to form a vector S_2 as follows:

$$S_2(i) \equiv Par_n(i) + RND_i(p)\sigma_2 d_2(i),\ \ d_1 \equiv A_2 - A_3 \big/ |A_2 - A_3| \tag{9}$$

3.2 Selection Mechanism

The tournament selection mechanism [10] for solving multi-objective problems is selected. In order to avoid the complicated niche counts, we intend to calculate the "*average distance*" $\overline{\mathbf{D}}_i$ to find suitable non-dominated solutions to maintain the diversity of archive, the mechanism is changed as follows.

Details of the tournament selection: *a*. select two candidates $\boldsymbol{A}$ and $\boldsymbol{B}$ randomly from the neighbors; *b*. form a comparison set with the rest of the neighbors′ individuals, namely ***CSet***; *c*. compares ***CSet*** with individual $\boldsymbol{A}$ and $\boldsymbol{B}$ separately. If there is only an individual $\boldsymbol{A}$ (or $\boldsymbol{B}$) dominate ***CSet***, then the only candidate is $\boldsymbol{A}$ (or $\boldsymbol{B}$); if $\boldsymbol{A}$ (or $\boldsymbol{B}$) is dominated by ***CSe***t, while the other one does not, then the individual which is dominated by ***CSet*** will be selected to participate in the evolution of next generation; if both $\boldsymbol{A}$ and $\boldsymbol{B}$ dominate **CSet**, select one $\boldsymbol{A}$ (or $\boldsymbol{B}$) randomly as the candidate and update lattice; if ***CSet*** dominate both $\boldsymbol{A}$ and $\boldsymbol{B}$, calculate the "*average distance*" $\overline{\mathbf{D}}_i$ of each agent in ***CSet***. The "*average distance*" $\overline{\mathbf{D}}_i$ is defined as follow:

$$\overline{\mathrm{D}}_i = \sum_{\kappa=1}^{|CSet|-1} \mathrm{distance}_k \,/\, (|CSet|-1)\ ,\ i=(1,2,...,|CSet|) \tag{10}$$

where $\mathrm{distance}_k$ is the distance between i-th agent and k-th agent in the ***CSet***, $k \neq i$, $k=(1, 2, \ldots, |\boldsymbol{CSet}|-1)$. To maintain the diversity of the solutions, an agent with a larger $\overline{\mathbf{D}}_i$ will be chosen as a candidate to update population.

3.3 Self-learning Operator

In DMOAGA, this operator is used on agents in archive, it is equivalent to a small-scale local search algorithm, hoping to obtain more valuable information from these elites and make solutions more approximate to the PF.

Firstly, random perturb in a small area of vicinity of each agent which is living in the archive, and generate an approximate agent in phenotype space. As a result, the size of archive expanded double temporarily. The perturbation area of agent i is given as: a circular area with a small radius θ and i-centered. Secondly, temporary archive is non-dominated sorted and updated.

3.4 Implementation of DMOAGA

$\boldsymbol{L}^t$ represents the agent lattice in the t-th generation, $\boldsymbol{Best}^t$ is the Non-dominated set of elites among $\boldsymbol{L}^0, \boldsymbol{L}^1, \ldots, \boldsymbol{L}^t$. $\boldsymbol{P}_m$, $\boldsymbol{P}_c$ are the probabilities to perform the mutation and crossover operator respectively. Neighborhood non-dominated solutions are singled out firstly by tournament mechanism for initializing a global archive. Then perform neighborhood competition, mutation operator and neighborhood orthogonal crossover operator with the certain probability for each agent, and update archive. Finally, perform self learning on every agent in the archive, update archive as well. This final archive is regarded as the Pareto set of DMOAGA. It is worth noting that the archive is updated during the whole evolutionary process.

Step1: Initialize $\boldsymbol{L}^0$, update $\boldsymbol{Best}^0$, and $t \leftarrow 0$;
Step2: Perform the neighborhood competition operator on each agent in $\boldsymbol{L}^t$;
Step3: For each agent in $\boldsymbol{L}^t$ if $\boldsymbol{U}(0,1)<\boldsymbol{P}_m$, perform the mutation operator on it;
Step4: For each agent in $\boldsymbol{L}^t$ if $\boldsymbol{U}(0,1)<\boldsymbol{P}_c$, perform the crossover operator on it;
Step5: Perform the direction perturbation operator on each agent in L^t, obtaining L^{t+1};
Step6: Update $\boldsymbol{Best}^t$;
Step7: perform the self-learning operator on $\boldsymbol{Best}^t$, and then update $\boldsymbol{Best}^t$;
Step8: If termination criteria are reached, output $\boldsymbol{Best}^t$ and stop; otherwise $t \leftarrow t+1$, go to Step 2.

4 Experiments

In the experiments, the unconstrained benchmark functions ZDT1~4[11] and UF1~6[12], are used to validate the performance of DMOAGA. We use red dotted representing the sampling of true Pareto front, and black circle indicates the solutions come from DMOAGA.

A variety of performance evaluations have been proposed [17], we will look at two popular criteria. The first measure is to calculate the GD [13], which is the average distance from a set of solutions found by evolution to the global POS. The second measure is to calculate the IGD [14-15]. In order to get a good value for IGD (ideally zero), Pareto solutions need to cover all parts of the POS. The more closer to 0 the better GD and IGD, as we can see from the formula, if and only if both convergence and distribution of Pareto set are good, the values of these two indicators will be close to 0.

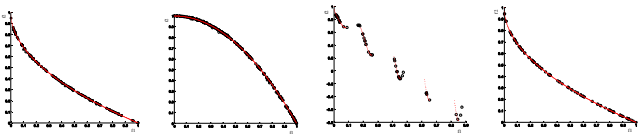
Table 1. Parameters of DMOAGA

Nvar	Number of variables	30	θ	Self-learning radius	0.05
Sgen	Number of self-learning generations	10	P_e	Probability of perturbation	0.4
P_m	Probability of mutation	0.1	*Obj*	Number of objectives	2
P_c	Probability of crossover	0.2	*gen*	Number of generations	10^3
P_o	Probability of competition strategy selection	0.2	L_{size}	Size of lattice	10

The parameters of DMOAGA are given in Table 1. $\boldsymbol{L}_{size}\times\boldsymbol{L}_{size}$ is equivalent to the population size in traditional GAs, so $\boldsymbol{L}_{size}$ can be selected from 5 to 10. The left five ones are chosen according to [4], θ controls the area of local search, so it is better to be a small value, P_e is same as [3], and *gen* equals to 1000 for the balance of evaluation and performance. Its program is run 30 times randomly. For unbiased comparison, NSGA-II and DMEA are also run 30 times respectively. The results are given in Table 2: these data show that DMOAGA have good performance about UF1~5 on GD as well as IGD except that the average value of UF5 on IGD is worse than the value of NSGA-II 0.0001. Even more, DMOAGA is much better than NSGA-II on most data of IGD. Fig.2 (a)~(d) show the PF (the red curve) and Pareto solutions obtained from DMOAGA (the black circle) on benchmark ZDT1~4, Fig.3 (a)~(f) show the PF (the red curve) and Pareto solutions obtained from DMOAGA (the black circle) on benchmark UF1~6. As we can see DMOAGA obtained a large number of satisfactory non-dominated solutions, and Pareto set almost accurately approximated and covered PF. It is proved that all of the innovations in this article are effective.

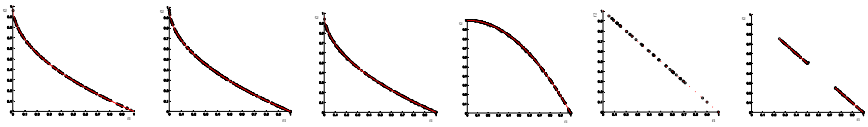
Table 2. Comparison among DMOAGA, DMEA, and NSGA-II in terms of GD and IGD

Problem	DMOAGA $\times10^{-4}$				DMEA $\times10^{-4}$				NSGA-II $\times10^{-4}$			
	GD		IGD		GD		IGD		GD		IGD	
	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD
UF1	**0**	0	**0**	0	-	-	-	-	0	1	144	100
UF2	**0**	0	**0**	0	-	-	-	-	0	0	120	100
UF3	**0**	0	**0**	0	-	-	-	-	1	36	128	90
UF4	**0**	0	**0**	0	-	-	-	-	0	0	141	110
UF5	**10**	0	40	0	-	-	-	-	23	20	45	40
UF6	12	1	10	0	-	-	-	-	5	2	3	0
ZDT1	8	0	**0**	0	3	0	51	3	6	0	47	100
ZDT2	12	20	**0**	0	3	0	42	1	4	0	48	110
ZDT3	145	19	**70**	10	4	0	108	10	5	0	74	10
ZDT4	323	0	80	0	5	3	49	2	9	0	48	10



(a) ZDT1 (b) ZDT2 (c) ZDT3 (d) ZDT4

Fig. 2. The solutions of DMOAGA on ZDT benchmark



(a) UF1 (b) UF2 (c) UF3 (d) UF4 (e) UF5 (f) UF6

Fig. 3. The solutions of DMOAGA on UF benchmark

5 Conclusion

This paper introduced a novel algorithm that employs directions information of archive and multi-agent systems to solve MOPs. Both of them complement each other, effectively found non-dominated solution set in the evolutionary process, and with good results on the approximation of the true PF. First of all, agents in a lattice exchange information with others using their own perception of the environment, competition or cooperation, and depth of local search using self-learning; secondly, the "*average distance*" rule is incorporated into the tournament selection mechanism to ensure that diversity of the evolutionary process; finally, at each generation, two types of directions are generated, with these two directions, give full play to the role of the archive information in finding POS, and reflect the nature of the agent which can cooperate with each other.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China under Grants 61271301 and 61103119, and the Fundamental Research Funds for the Central Universities under Grant K5051202052.

References

1. Liu, J., Zhong, W., Jiao, L.: Moving Block Sequence and Organizational Evolutionary Algorithm for General Floor Planning with Arbitrarily Shaped Rectilinear Blocks. IEEE Trans. on Evolutionary Computation 12(5), 630–646 (2008)
2. Jiao, L., Liu, J., Zhong, W.: An Organizational Co-evolutionary Algorithm for Classification. IEEE Trans. on Evolutionary Computation 10(1), 67–80 (2006)
3. Bui, L.T., Liu, J., Bender, A., Barlow, M., Wesolkowski, S., Abbass, H.A.: DMEA: A Direction-based Multi-objective Evolutionary Algorithm. Memetic. Comp. 3(4), 3:271-3:285 (2011)

4. Zhong, W., Liu, J., Xue, M., Jiao, L.: A Multi-agent Genetic Algorithm for Global Numerical Optimization. IEEE Trans. on Systems, Man and Cybernetics 34(2), 1128–1141 (2004)
5. Liu, J., Zhong, W., Jiao, L.: A Multi-agent Evolutionary Algorithm for Combinatorial Optimization Problems. IEEE Trans. on Systems, Man, and Cybernetics Part B 40(1), 229–240 (2010)
6. Liu, J., Jing, H., Tang, Y.Y.: Multi-agent Oriented Constraint Satisfaction. Artif. Intell. 136(1), 101–144 (2002)
7. Liu, J.: Autonomous Agents and Multi-Agent Systems: Explorations in Learning Self-Organization, and Adaptive Computation. World Scientific, Singapore (2001)
8. Pan, Z.J., Kang, L.S., Yao, X., Kim, J.H., Furuhashi, T.: An Adaptive Evolutionary Algorithms for Numerical Optimization. In: Kim, J.-H., Furuhashi, T., Yao, X. (eds.) SEAL 1996. LNCS (LNAI), vol. 1285, pp. 27–34. Springer, Heidelberg (1997)
9. Leung, Y.W., Wang, Y.: An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. IEEE Trans. Evol. Comput. 5, 41–53 (2001)
10. Horn, J.R., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation, vol. 1, pp. 82–87. IEEE Service Center, Piscataway (1994)
11. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multi-objective Evolutionary Algorithms: Empirical Results. Evol. Comput. 8(2), 173–195 (2000)
12. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multi-objective Optimization Test Instances for The CEC 2009 Special Session and Competition, Technical Report CES-487, The School of Computer Science and Electronic Engineering, University of Essex, Tech. Rep (2008)
13. Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary Computation and Convergence to a Pareto Front. In: Koza, J.R. (ed.) Late Breaking Papers at the Genetic Programming 1998 Conference, pp. 221–228. Stanford University, Stanford University Bookstore, California (1998)
14. Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A Regularity Model-based Multi-objective Estimation of Distribution Algorithm. IEEE Trans. Evol. Comput. 21(1), 41–63 (2008)
15. Sierra, M.R., Coello, C.A.C.: A Study of Fitness Inheritance and Approximation Techniques for Multi-objective Particle Swarm Optimization. In: Proc. Congr. Evol. Comput (CEC 2005), pp. 65–72 (2005)
16. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multi-objective Optimization. In: Giannakoglou, K.C., Tsahalis, D.T., Periaux, J., Papailiou, K.D., Fogarty, T. (eds.) Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems. International Center for Numerical Methods in Engineering (Cmine), pp. 95–100 (2001)
17. Coello, C.A.C, Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-objective Problems. Kluwer (2002)

A Study of Representations for Resource Constrained Project Scheduling Problems Using Fitness Distance Correlation

Bingqi Cai and Jing Liu

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China
caibingqi1208@126.com, neouma@mail.xidian.edu.cn

Abstract. Four representations for resource constrained project scheduling problems (RCPSPs) are studied by making use of the fitness landscape analysis technique. The fitness distance correlation (FDC) measure is used to analyze the landscapes. In the experiments, the study on the benchmark problems J30 is first presented to investigate which distance metric is more suitable for calculating FDC for RCPSPs. Then, the benchmark problems Patterson, J30, and J60 are used to evaluate the effect of the four encodings on the performance of evolutionary algorithms. Finally, a standard genetic algorithm is applied to verify the predictions made by the FDC. To the best of our knowledge, this is the first work on using FDC to study different encodings for RCPSPs.

Keywords: Fitness landscape analysis, Fitness distance correlation, Representation, Resource constrained project scheduling problems.

1 Introduction

Evolutionary algorithms (EAs) have been successfully used to solve various problems [1-5]. The search efficiency of an EA is affected by the representation and associated variation operators. Therefore, designing suitable encodings and variation operators are key issues in improving EAs' performance.

The fitness landscape analysis [6] has been shown to be a valuable tool in investigating how a given EA works. Thus, the choice of an encoding and operators can be made based on the study of the difficulties of the corresponding fitness landscapes. Many fitness landscape analysis measures have been proposed, such as FDC [7], motif difficulty [8], and correlation length [9]. The FDC is one of the most popular measures and has been employed to investigate different problems [9-10].

Resource constrained project scheduling problems (RCPSPs) are a kind of well-known combinatorial optimization problems, and exist in many practical engineering applications, such as planning, management, and production. The objective is to find a schedule with minimum makespan T observing both precedence relations and resource limits. Many EAs have been proposed to solve RCPSPs [11-13]. For this problem, there are many different representations. Thus, it is essential to study the effect of representations carefully for the design of an effective algorithm. However, there is very few work on studying the problem.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 218–225, 2013.

In this paper, we used the fitness landscape framework to study different encodings for RCPSPs. The FDC measure is used to analyze fitness landscapes. A computational study on J30 is presented to investigate which distance metric is more suitable for calculating FDC. Then, we study the FDC of three instance sets to predict the impact of the four encodings on the performance of EAs. Finally, a standard genetic algorithm is applied to verify the predictions made by the FDC. To the best of our knowledge, this is the first work on using FDC to study different encodings for RCPSPs. Czogalla et al. [10] also studied fitness landscapes for RCPSPs, but they did not analyze the effect of different encodings.

The rest of this paper is organized as follows. Section 2 introduces the model of RCPSPs. In Section 3, existing evolutionary techniques applied to RCPSPs are described. Section 4 presents the concept of fitness landscapes and the analysis techniques used in this work. Experimental results and analysis are given in Section 5, and the conclusions are given in Section 6.

2 Resource Constrained Project Scheduling Problems

In a RCPSP, the project consists of a set J of n activities, $J = \{1, 2, \ldots, n\}$ and a set K of k renewable resources, $K = \{1, 2, \ldots, k\}$, each resource having a limited capacity R_k. The duration of activity $j \in J$ is d_j. Generally, two tasks, 1 and n, are added, corresponding to two fictitious activities of zero duration. Each activity j requires r_{jk} units of resource k during each period of its duration d_j. The structure of a project can be depicted by an activity-on-node (AON) network $G = (V, A)$, where V is the set of activities and A is the set of precedence relationships [12]. Throughout this paper, we denote the set of immediate predecessors of activity j by P_j. S_j is the set of the immediate successors of activity j and $\overline{S_j}$ is the set of all direct and indirect successors of j. Formally, this is stated as follows:

$$\text{Minimize} \quad c_n \tag{1}$$

$$\text{Subject to} \quad s_j \geq \max_{h \in P_j} c_h \quad \forall j \in J \tag{2}$$

$$\sum_{j \in A_t} r_{jk} \leq R_k \quad \forall k \in K, t \in T \tag{3}$$

where s_j indicates the start time of activity j and c_h indicates the finish time of activity h. A_t denotes the set of activities which are being processed at the time of t.

3 Evolutionary Algorithms for RCPSPs

EAs have been in wide use for solving RCPSPs. There are many choices of encodings and operators. This indicates that a careful study of representations and operators is crucial for the design of an effective EA. In this work, the most ordinary representations for this problem are studied.

3.1 Random Activity List Representation

The random activity list representation (*Randlist*) is a typical encoding and has been widely applied to the RCPSP [11], [14]. This encoding is given by an activity sequence, which is assumed to be a precedence feasible permutation of the set of activities. It is computed as follows: Starting with the empty activity list, we gain a precedence feasible activity list by repeatedly applying the following step: The next activity in the activity list is randomly taken from the set of those currently unselected activities all non-dummy predecessors of which have already been chosen for the activity list. The encoding is formed indirectly, so that there is a need for a method to derive a schedule from a precedence feasible activity list. Each activity list is related to a uniquely determined schedule which is decoded by the serial schedule generation scheme [14]. This encoding calls for genetic operators that can preserve the permutations and precedence relationships between activities, such as the order-based two-point crossover operator and insert mutation operator [12].

3.2 Activity List Representation Based on the Latest Finish Time Rule

Activity list representation based on the latest finish time rule (*LFTlist*) is a variant of *Randlist*, which determines the activity list with a sampling procedure as discussed by Kolisch [14]. More precisely, a priority rule is employed to derive probabilities which are used to select the next activity for the activity list. In this paper, we adopt the latest finish time rule (LFT), which is a better priority rule in comparison with some other priority rules, such as the most total successors rule (MTS). Decoding the activity list into a feasible schedule is done by the same manner as in the *Randlist*. The variation operators applied are also the same as the *Randlist*.

3.3 Priority Rule Representation

The priority rule representation (*Prule*) has been developed by Dorndorf and Pesch [15]. In this representation, a chromosome I consists of a sequence of priority rules

$$I = (pr_1, pr_2, \ldots, pr_n) \tag{4}$$

where each position $i = 1, 2, \ldots, n$ is associated with a priority rule

$$pr_i \in \{\text{LFT, LST, MTS, WRUP, GRPW}\}. \tag{5}$$

The five priority rules used in this study are selected based on reported results in previous studies [11], [14]. These priority rules are among the best reported priority rules in the literatures. This representation allows the use of the classical crossover and mutation operators. We use a serial schedule generation scheme as the one applied in the previous section to decode the representation.

3.4 Random Key Representation

In the random key representation (*Randkey*) [11], a chromosome I is represented by a vector of real values $I = (w_1, w_2, \ldots, w_n)$, which is determined by randomly drawing a real value $w_j \in [0, 1]$ with a uniform distribution for each activity. Classical

crossover and mutation operators are applied to this representation. We also use a serial schedule generation scheme to decode this representation.

4 Fitness Landscapes and Fitness Distance Correlation

The metaphor of a fitness landscape is commonly used for descriptions of the application of EAs for solving a combinatorial optimization problem. Moreover, the study of fitness landscapes can be valuable in designing EAs since it can help predict EAs' performance. The fitness landscape is composed of three main components: a set of genotypes (solutions) S, a fitness function f which assigns a real-valued fitness to each solution in S and a distance function d defining neighborhood relations between the solutions. For details of fitness landscapes for EAs, refer to [6].

FDC is one of the most commonly used landscape analysis techniques. It is proposed by Jones and Forrest [7], and is one method of quantifying the relationship between fitness value and distance to the nearest optimum. Given a set of points $x_1, x_2, \ldots, x_m$, and two corresponding sets $F = \{f_1, f_2, \ldots, f_m\}$ and $D = \{d_1, d_2, \ldots, d_m\}$, the FDC coefficient r is defined as:

$$r = \frac{1}{\sigma_f \sigma_d m} \sum_{i=1}^{m} (f_i - \overline{f})(d_i - \overline{d}) \tag{6}$$

where f_i is the fitness value, d_i is the minimum distance to a global optimum solution, and $\overline{f}$, $\overline{d}$, σ_f and σ_d are the means and standard deviations of F and D respectively. The FDC coefficient can be estimated using randomly generated samples in the search space.

5 RCPSP Landscape Analysis

5.1 Experimental Setup

Two sets of benchmark problems have been used. One is the Patterson set assembled by Patterson [16], and the other is generated by a project generator ProGen developed by Kolisch et al. [17]. In our study, we have used the ProGen problem instances with 30 and 60 non-dummy activities, namely J30 and J60. In order to simplify the experiment, we randomly select some simple instances from Patterson, J30, and J60, and the number is 23, 19, and 14 respectively.

There is some redundancy in the search space since distinct genotypes may be related to the same schedule. In addition, different schedules might possess the same makespan. Therefore, there are multiple optimal solutions in the search space. However, the global optima are required for the FDC analysis and it is not easy to find so many optimal solutions in the genotype space. The optimal solutions in the phenotype space are less relatively. Thus, we calculate the distance between the optimal solutions and the candidate solutions at the phenotype level. This simply means that all solutions contained are converted to its phenotype, namely a vector of the start times of activities. The instances chosen in this paper are relatively simple, so

we can obtain the global optima of these instances by an EA [12]. We assume that the algorithm stops when the maximum number of fitness function evaluations is achieved and the number is set as 10,000. For each instance, the algorithm is run at most 30 times and stops as soon as 10 distinct optimal solutions are found.

In the following experiments, FDC is calculated exhaustively when the instance search space contains 4,000 or fewer solutions, and it is estimated by a sample of 4,000 randomly chosen solutions in the search space otherwise. The fitness of a solution I is calculated as follows: fitness (I) = -makespan (I).

5.2 Experiments on Different Distance Metrics

In experiments designed to discover the best distance metric, J30 set is used, from which we select 19 instances. Due to the randomness, the calculation of FDC for each instance is repeated 20 times, and the average is taken over the 19 instances. The experimental results are given in Table 1.

The result shows that based on the Euclidean distance the FDC value of the *randlist* and *randkey* representations is lower than that of the *LFTlist* and *Prule* representations. The FDC value based on the Manhattan distance is similar to that obtained by the Euclidean distance. For the result based on the Hamming distance, the FDC value of the *LFTlist* and *Prule* representations is lower than that of the *randlist* and *randkey* representations.

Table 1. Average Values of FDC based on different distance metrics on J30 Set

r	Euclidean distance	Manhattan distance	Hamming distance
LFTlist	-0.76	-0.77	**-0.44**
Prule	-0.72	-0.72	-0.43
randlist	-0.84	-0.79	-0.30
randkey	**-0.86**	**-0.82**	-0.29

In the following study, we will test whether the prediction made by FDC is consistent with the results obtained by evolutionary optimization. The standard GA with the four encodings is used to solve the 19 instances from J30. Two-point crossover operator is applied to the four GA variants and the crossover rate is 0.75. This crossover operator is different from traditional two-point crossover operator. For details of this operator, refer to [12]. The insert mutation operator [12] is used to the *LFTlist* and *randlist* GA variants. This mutation operation is not too disruptive and it maintains the inherent order imposed by the precedence relations, so in this paper we set the mutation rate as 0.9 to improve diversity. The *Prule* and *randkey* GA variants apply the positional mutation operator [11] and the mutation rate is 0.05.

In addition, an elitist-preserving approach is adopted. The best individual is always kept in the population. After a number of trials we set the population size as 40 and the maximum number of fitness function evaluations as 2,000. The algorithm stops as soon as the optimal solution is found or the maximum number of fitness function evaluations is achieved. The solution quality is evaluated by the *Error*

$$Error = \frac{solution\ makespan - optimal\ makespan}{optimal\ makespan} 100\% . \quad (7)$$

Table 2 summarizes the percentage of average error (%AvgErr) over 20 independent runs per instance over the 19 instances. In general, the GAs with the *LFTlist* and *Prule* representations obtain the best results, which is consistent with the prediction made by the FDC based on the Hamming distance. Therefore, we will use the Hamming distance in the rest of the experiments.

Table 2. The Percentage of Average Error on J30 Set

GA variants	*GA-LFTlist*	*GA-Prule*	*GA-randlist*	*GA-randkey*
%AvgErr	**1.07**	1.29	1.55	1.46

5.3 Experiments on FDC Based on the Hamming Distance

In this experiment, the influence of the four encodings on EAs for RCPSPs is studied and FDC is calculated by the Hamming distance. For each chosen instance, FDC is calculated 20 times. The average is taken over the chosen instances over all replications. Table 3 summarizes the average value of FDC. The values in bold imply the best representation for a given set.

From Table 3, we can see that the *LFTlist* representation has the lowest FDC value for all the three sets. This suggests the *LFTlist* representation is likely to make search easier than the three other encodings. For Patterson set, the *Randlist* representation has a second lowest FDC value and the two other encodings have higher FDC values. The *Prule* encoding has a second lowest FDC value for J30 set and J60 set.

Table 3. Average Values of FDC on Different Test Sets

Representation	Instance Set	*r*
LFTlist	Patterson set	**-0.35**
Prule	Patterson set	-0.25
Randlist	Patterson set	-0.33
Randkey	Patterson set	-0.32
LFTlist	J30 set	**-0.44**
Prule	J30 set	-0.43
Randlist	J30 set	-0.30
Randkey	J30 set	-0.29
LFTlist	J60 set	**-0.51**
Prule	J60 set	-0.44
Randlist	J60 set	-0.35
Randkey	J60 set	-0.34

The landscape analysis provides some insights about the behavior of these different genetic representations during an evolutionary process. Nevertheless, it is crucial to relate this with results obtained by optimization runs. Therefore, we apply four GA variants using the four different encodings to the RCPSP to verify whether the prediction made by FDC is in accordance with the evolutionary optimization results.

The parameter setting is the same as that in Section 5.2 except the population size and maximum number of fitness function evaluations of Patterson set and J60 set. The number of activities of the three sets is different and by a number of trials we find the best population size and appropriate maximum number of fitness function evaluations for these sets are also different. For Patterson set, the population size is the number of activities in each instance and the maximum number of fitness function evaluations is 1,000. For J60 set, the population size is 70 and the maximum number of fitness function evaluations is 5,000. Table 4 displays the %AvgErr over 20 independent runs on the chosen instances.

In Table 4, it can be seen that the *LFTlist* representation is the best approach, followed by the *Randlist* representation for the Patterson set and the *Prule* representation for J30 set and J60 set. These optimization results are consistent with the results of previous fitness landscape analysis.

Table 4. The Percentage of Average Error on Different Test Sets

Instance Set	%AvgErr			
	LFTlist	*Prule*	*Randlist*	*Randkey*
Patterson set	**0.78**	2.05	1.42	2.45
J30 set	**1.07**	1.29	1.55	1.46
J60 set	**0.09**	0.36	0.79	0.74

6 Conclusion

A fitness landscape analysis has been presented for the RCPSP in this paper. FDC, a popular measure for fitness landscape analysis is used to investigate how encodings influence the search performance of EAs for the RCPSP. There is very few study on analyzing this problem though it is an important issue.

Four representations are examined. We use the FDC to predict the performance of these encodings and subsequent evolutionary optimization results verify the prediction. The results show that the activity list representation based on the latest finish time rule performs best.

The choice of a suitable representation plays a crucial role on the design of an effective EA. Hence, the analysis of different representations in this paper may be useful as helpful guidelines in designing other EAs to solve the RCPSP.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China under Grants 61271301 and 61103119, and the Fundamental Research Funds for the Central Universities under Grant K5051202052.

References

1. Zhong, W., Liu, J., Xue, X., Jiao, L.C.: A Multiagent Genetic Algorithm for Global Numerical Optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34, 1128–1141 (2004)

2. Liu, J., Zhong, W., Jiao, L.C.: A Multiagent Evolutionary Algorithm for Constraint Satisfaction Problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36, 54–73 (2006)
3. Liu, J., Zhong, W., Jiao, L.C.: Moving Block Sequence and Organizational Evolutionary Algorithm for General Floorplanning with Arbitrarily Shaped Rectilinear Blocks. IEEE Transactions on Evolutionary Computation 12, 630–646 (2008)
4. Jiao, L.C., Liu, J., Zhong, W.: An Organizational Coevolutionary Algorithm for Classification. IEEE Transactions on Evolutionary Computation 10, 67–80 (2006)
5. Liu, J., Zhong, W., Jiao, L.C.: A Multiagent Evolutionary Algorithm for Combinatorial Optimization Problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B 40, 229–240 (2010)
6. Jones, T.: Evolutionary Algorithms, Fitness Landscapes and Search. Ph.D. dissertation, University of New Mexico, Albuquerque, New Mexico (1995)
7. Jones, T., Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 184–192 (1995)
8. Liu, J., Abbass, H.A., Green, D.G., Zhong, W.: Motif difficulty (MD): A Predictive Measure of Problem Difficulty for Evolutionary Algorithms using Network Motifs. Evolutionary Computation Journal 20, 321–347 (2012)
9. Tavares, J., Pereira, F.B., Costa, E.: Multidimensional Knapsack Problem: A Fitness Landscape Analysis. IEEE Transaction on Systems, Man, and Cybernetics 38, 604–616 (2008)
10. Czogalla, J., Fink, A.: Fitness Landscape Analysis for the Resource Constrained Project Scheduling Problem. In: Stützle, T. (ed.) LION 3. LNCS, vol. 5851, pp. 104–118. Springer, Heidelberg (2009)
11. Hartmann, S.: A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. Naval Research Logistics 45, 733–750 (1998)
12. Hindi, K.S., Yang, H., Fleszar, K.: An Evolutionary Algorithm for Resource-Constrained Project Scheduling. IEEE Transactions on Evolutionary Computation 6, 512–518 (2002)
13. Kolisch, R., Hartmann, S.: Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. European Journal of Operational Research 174, 23–37 (2006)
14. Kolisch, R.: Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. European Journal of Operational Research 90, 320–333 (1996)
15. Dorndorf, U., Pesch, E.: Evolution based Learning in a Job Shop Scheduling Environment. Computers & Operations Research 22, 25–40 (1995)
16. Patterson, J.H.: A Comparison of Exact Approaches for Solving the Multiple Constrained Resource Project Scheduling Problem. Management Science 30, 854–867 (1984)
17. Kolisch, R., Sprecher, A., Drexl, A.: Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. Management Science 41, 1693–1703 (1995)

Adapt a Text-Oriented Chunker for Oral Data: How Much Manual Effort Is Necessary?

Isabelle Tellier[1], Yoann Dupont[1], Iris Eshkol[2], and Ilaine Wang[1]

[1] Lattice, University Paris 3 - Sorbonne Nouvelle
[2] LLL, University of Orléans

Abstract. In this paper, we try three distinct approaches to chunk transcribed oral data with labeling tools learnt from a corpus of written texts. The purpose is to reach the best possible results with the least possible manual correction or re-learning effort.

1 Introduction

The annotation of transcribed spontaneous speech is a difficult task, because oral corpora are full of irregularities and *disfluencies*. In this paper, we are mainly interested in the task of chunking transcribed oral data. A "chunk" is a non-recursive constituent of linguistic units [Abn91]. The purpose of a chunker is thus to identify the sentence constituents without specifying their internal structure and their syntactic function. This analysis relies on POS labels and can be considered as the best possible that can be reached for such data, for which it is not always possible to provide a full syntactic parsing.

To build a POS tagger and a chunker, several strategies can be considered. The main two options are either writing rules by hand or using supervised machine learning techniques on labeled data. We favor the machine learning approach, which requires less effort and performs better. Unfortunately, for many languages (this is the case for French, the language we are interested in), annotated transcribed oral corpora are rare. We are in a situation where a large corpus of fully annotated written sentences is available, whereas only a small corpus of annotated transcribed oral data is. Our annotated oral data are not large enough to learn a specific POS tagger, but learning a chunker requires less data. Is it worth doing it? The problems we address in this paper are thus the following: to chunk transcribed data, is it better to use a chunker learnt from a large corpus of written texts or one learnt from a small but specific sample of the target transcribed corpus? How much improvement can a limited manual effort bring? To address these questions, we propose three distinct protocols with increasing amount of human intervention, and compare their effectiveness.

The first section of this paper describes the chunking task, its specificities for transcribed oral data and the machine learning technique used. The second section is dedicated to the labeled corpora at our disposal: a large corpus of written texts and a small corpus of transcribed oral data, and their corrected versions. In the last section, we propose three strategies to chunk the transcribed data with various levels of manual adaptation and compare their results.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 226–233, 2013.

2 The Chunking Task

2.1 Chunking Transcribed Oral Data

Chunkers, also called shallow parsers, are well adapted for transcribed oral data in which "sentences" are not fully syntactically correct. Some software tools provide this type of analysis but they usually don't behave well on oral data. The reasons are the lack of punctuation marks and the *disfluencies*, which are standardly seen as positions in the speech flow where the linearity is broken. They are very numerous in spoken texts and of various types: repetitions (e.g. "la la" in French, i.e. "the the"); immediate self-corrections (e.g. "le la", i.e. "the (masculine) the (feminine)"); false start (e.g. "les dans" i.e. "the in"), word fragments (e.g. "vous v- vous", i.e."you y- you").

Some attempts to build a chunker specifically adapted to French transcribed data have been developed in France. Most of them consisted in iteratively applying hand-written finite-state transducers, together with lexical and syntactic resources. [BCDW10] tried to automatically annotate French corpora of spontaneous speech transcriptions in super-chunks, i.e. chunks containing complex multiword units. Their parsing was based on a preprocessing stage of the spoken data consisting in reformatting and tagging utterances containing disfluencies. A similar approach has been conducted in [VV99] for POS tagging. [AMF08] proposed another strategy by including a post-correction stage in order to deal with chunking errors due to disfluencies.

Our approach is different. Following [BB05], we believe that disfluence phenomena should be included in the analysis of language even if it raises specific processing issues. To deal with real data and avoid *ad hoc* handmade programs, we favor a machine learning strategy. We detail in the following the kind of learning models we use. A similar strategy has already been applied for POS labeling of French transcribed oral data in [TETP10], but it was based on a different tagset than the one used here.

2.2 Machine Learning for POS Labeling and Chunking

To perform both POS labeling and chunking, we used the state of the art machine learning approach for annotation tasks: Conditional Random Fields (or CRFs). As it has often been observed [SP03, CT12] they behave very well on this task.

Introduced in [LMP01], CRFs belong to the family of graphical models. When the graph is linear (which is most often the case), the probability distribution that the annotation sequence y is associated with the input sequence x is:

$$p(y|x) = \frac{1}{Z(x)} \prod_t \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, x) \Big)$$

Where $Z(x)$ is a normalization factor. This computation is based on K features f_k (usually binary functions), provided by the user. The feature f_k is activated (i.e. $f_k(t, y_t, y_{t-1}, x) = 1$) if a configuration occurring at the current position t

in the sequence, concerning y_t, y_{t-1} (i. e. the values of the annotation at the positions t and $t-1$) and x is observed. Each feature f_k is associated with a weight λ_k, estimated during the learning step. The most efficient implementation of linear CRFs is Wapiti[1], which uses L1 and L2 penalizations to select the best features during the learning step [LCY10]. It is the software we have used.

3 Corpora and Labeling Conventions

In this section, we first describe two French corpora at our disposal. The first one, called FTB (French TreeBank), is a treebank of written sentences, which can thus be easily transformed into a labeled corpus for POS and chunking. The second one, called ESLO 1 (Enquête Sociolinguistique d'Orléans[2]), is made of transcribed oral data. Originally, it was not at all labeled.

Our basic methodology consists in applying labeling tools learnt from the corpus of written sentences to the corpus of transcribed oral data. To measure the effectiveness of these tools, we had to build a reference labeled version of the transcribed corpus manually. Some linguistic choices had thus to be made to take into account the specificities of our transcribed data while remaining as much as possible compatible with the labels of the written sentences. To do so, we defined some labeling conventions, especially concerning the disfluencies. In fact, we produced two reference labeled versions of the ESLO 1 corpus tagged with the labels of the FTB, with an increasing level of adaptation. We describe these two variants (corpus 1 and corpus 2) in the last two subsections.

3.1 French TreeBank (FTB)

The variant of the FTB we used is made of about 10 000 fully parsed sentences extracted from articles of the newspaper "Le Monde" [ACT03]. The set of 30 POS tags is described in [CC08]. The distinct possible kinds of chunks, together with the possible POS tags corresponding to their head, are the followings: AP for adjectival chunk (ADJ, ADJWH), AdP for adverbial chunk (ADV, ADVWH, I for interjection), CONJ for conjunction chunk (CC, CS), NP for nominal chunk (CLO, CLR, CLS, NC, NPP, PRO, PROREL, PROWH), PP for prepositional chunk (P, P+D, P+PRO), VN for verbal chunk (V, VIMP, VINF, VPP, VPR, VS). To transform a chunk analysis into a word annotation, we use the classical BIO labeling format (B for Beginning, I for In, O for Out is useless here since every word is member of a chunk).

3.2 ESLO 1

The ESLO 1 campaign gathered a large oral corpus among which we extracted for our experiments a sub-corpus of 8093 graphical words (i.e. tokens between two separators) belonging to 852 speaking turns. The main principles of our

[1] `http://wapiti.limsi.fr/`
[2] Sociolinguistic Survey of Orléans.

transcription guidelines are those followed by the researchers of the domain. First, words are transcribed using their standard spelling. Transcriptions do not contain any punctuation marks because the notion of sentence is not considered relevant [BBJ87]. The texts of ESLO 1 display disfluencies phenomena that are specific to spoken language: examples of repetitions, self-corrections, truncations, etc. it contains are provided in the following.

3.3 Labeling ESLO 1 with the FTB Labels: Corpus 1

We describe here how we choose to label oral disfluencies.

Repetition: When a token is repeated, the POS tags of both tokens are the same but, at the chunk level, two cases are possible:
• if the repeated token is a chunk head, then two distinct chunks are defined: (et/CC)$_{CONJ}$ (et/CC)$_{CONJ}$ (elle/CLS)$_{NP}$ (me/CLO)$_{NP}$ (disait/V)$_{VN}$[3]
• in every other case, both tokens belong to the same chunk:
(la/DET la/DET jeune/ADJ fille/NC)$_{NP}$[4]

Discourse markers are considered as interjections (I) and put into an adverbial chunk: (on/CLS)$_{NP}$ (peut/V)$_{VN}$ (commencer/VINF)$_{VN}$ (bon/I)$_{AdP}$ (alors/I)$_{AdP}$[5]

False starts and word fragments which are impossible to interpret are labelled as interjections and are part of an adverbial chunk: (c'/CLS)$_{NP}$ (est/V)$_{VN}$ (difficile/ADJ)$_{AP}$ (heu/I)$_{AdP}$ (les/I)$_{AdP}$ (dans/P ma/DET classe/NC)$_{PP}$[6]

Others interruptions of the morpheme being enunciated are interpreted according to the context:
• (vous/PRO)$_{NP}$ (êtes/V)$_{VN}$ (in-/NC)$_{NP}$ (institutrice/NC)$_{NP}$[7]
• (chez/P vous/PRO)$_{PP}$ (chez/P v-/PRO)$_{PP}$[8]

3.4 An Adapted Version of the Labeling for Oral Data: Corpus 2

To go further, it is also possible to define new specific chunks to treat disfluencies. To build the corresponding reference corpus, we made some new choices.

UNKNOWN POS Tag and Chunk: The UNKNOWN tag exists in the FTB, where it is assigned to foreign words. In corpus 2, we choose to also assign it to false starts, word fragments and orthographic errors of transcriptors. It is thus both a POS tag and a new kind of chunk.

[3] and and she told me
[4] the the young girl
[5] we can start well then
[6] it is difficult er the in my classroom
[7] you are schoo- schoolteacher
[8] at your's at y-

Interjection chunk (IntP) is also added as a new kind of chunks, used for interjection phrases and discourse markers: (c'/CLS)$_{NP}$ (est/V)$_{VN}$ (difficile/ADJ)$_{AP}$ (euh/I)$_{IntP}$ (les/UNKNOWN)$_{UNKNOWN}$ (dans/P ma/DET classe/NC)$_{PP}$[9]
Interjections and discourse markers can be part of a nominal chunk in the case of hesitations inside a NP like in:
• (l'/DET école/NC euh/I publique/ADJ)$_{NP}$[10]
• (des/DET hm/I inconvénients/NC)$_{NP}$[11]
unlike those *following* the nominal chunks: (des/DET idées/NC laïques/ADJ)$_{NP}$ (quoi/I)$_{IntP}$[12]

4 Three Experiments

We now describe in details three distinct experiments we have conducted to label our transcribed oral data and the results obtained. These experiments require an increasing amount of manual effort. In the following, the evaluation of the chunks is done using the strict equality criterion, meaning that two chunks are considered equal if and only if they share exactly the same frontiers and type.

4.1 First Approach: Direct Use of Written Texts Oriented Tools

The first and most simple strategy consists in applying the POS tagger and the chunker learnt from the FTB in cascade, without any adaptation, on the transcribed oral data of ESLO 1 (with corpus 1 as reference labelled data).

Figure 1 shows the templates used to define the features of the CRF for the POS tagging and for the chunking. For the POS tagging (on the left), we also used an external resource called the LeFFF (Lexique des Formes Fléchies du Français[13]) which is integrated into the CRF as a set of boolean attributes, one per distinct POS tag, representing whether or not the word is associated to this specific tag in the LeFFF. For the chunker (on the right) the features of the CRF are based on the correct POS tags of the FTB.

Figure 2 shows the results of these first experiments. The evaluations on the FTB are made by a 10-fold cross-validation. We evaluate the chunking with the micro- (resp. macro-) average of the F-measures of the obtained chunks, which is the average of the F-measures of every kind of chunk weighted (resp. not weighted) by their frequencies. Note that for corpus 1, the chunker is applied in cascade after the POS tagger, whereas it is based on the correct POS tags for the FTB. As expected, for oral data the quality loss is important on POS tagging (nearly 17%) and even worse when cascading the chunker afterwards. These poor results justify manual effort to adapt the tools.

[9] it is difficult er the in my classe
[10] the public er school
[11] some hm disadvantages
[12] laicist ideas what
[13] French Inflected Forms Lexicon

Feature	Type on y	window on x
Starts with upper ?	Unigram	[-2 .. 1]
Is punctuation ?	Unigram	[-2 .. 1]
Is a decimal ?	Unigram	[-2 .. 1]
3 last word's letters	Unigram	[-2 .. 1]
LeFFF information	Unigram	[-2 .. 1]
positional annotation	Bigram	$\emptyset$

Feature	Type on y	Window on x
Word	Unigram	[-2 .. 1]
POS	Bigram	[-2 .. 1]

Fig. 1. Feature Templates for POS and Chunk Learning resp. on the FTB

corpus	accuracy of the POS	micro-average of the chunker	macro-average of the chunker
FTB	97,33%	97,53	90,4
corpus 1	80,98%	77,24	76

Fig. 2. Results of the First Approach

4.2 Second Approach: Manual Correction of the POS Labeling

The second approach integrates a manual intervention after the application of the POS tagger, to correct the tags assigned to the transcribed corpus. The chunker learnt from the FTB is then applied on this corrected version of the oral data. This process is displayed on Figure 3, the manual effort being in bold. This correction is meant to compensate errors made by the POS tagger which typically depend on differences between written and oral productions. For example, "bon"[14] is used 99% as an adjective in the FTB, whereas it is much more frequently (83%) an interjection in corpus 1, "oui" and "non" [15] are labeled as ADV, I or NC in the FTB, while they are only interjections in corpus 1 (the reference corpus for this experiment). In this approach, the chunker is applied on correct POS tags for the oral data, as previously on the FTB. 1593 POS tags out of 8093 had to be manually corrected. The new micro-average of chunks on corpus 1 is then 87,74 while the new macro-average is 88,43. We have a significant improvement of results.

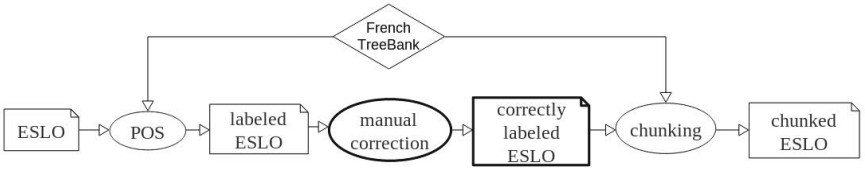


Fig. 3. Protocol of the Second Approach

[14] "well"

[15] "yes" and "no"

4.3 Third Approach: Learning of a Specific Chunker for Oral Texts

The last approach consists in learning a new chunker from the corrected transcribed speech data. We have already noted that learning a POS tagger would require a larger amount of data (i.e. a stronger manual labeling effort). The situation is different for a chunker, which mainly relies on the POS tags. The POS tags are much less numerous than words, and are thus much more likely to provide useful repetitions. Furthermore, this strategy gives the opportunity to define oral-specific chunks, as explained in section 3.4. The process used is displayed on Figure 4. The reference corpus for this experiment is thus corpus 2. The manual correction concerns 902 chunk tags out of 8093.

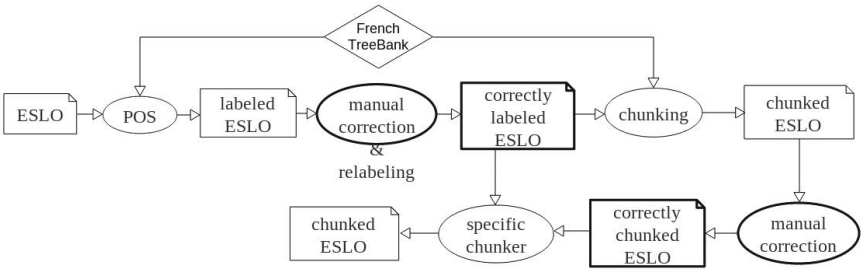


Fig. 4. Protocol of the third experiment

Table 5 shows the information used to train the CRF for the oral-specific chunk labeling task and the results obtained by 10-fold cross-validation.

Feature	Type on y	Window on x
Word	Unigram	[-2 .. 0]
POS	Unigram & Bigram	[-2 .. 1]
POS couple	Unigram	{-2,0} & {-1,0}

F-measures	on corpus 2
micro	96.65
macro	96.08

Fig. 5. Feature templates and F-measures on fully corrected ESLO

5 Observations and Conclusion

Analyzing the results, some observations can be made. First, the major improvements between the second and the third experiments concern adverbial (AdP) and nominal (NP) chunks. This is due to the high number of AdPs in the oral corpus, because of the interjections it contains. The situation is different for the conjunctive chunk (CONJ), equally well treated by both experiments. Finally, Adjectival (AP) and verbal chunks (VN) are better treated by the second experiment, probably because they are more frequent in written data.

Our oral corpus is characterized by a very significant number of interjections and discursive markers. The introduction of the new IntP chunk in corpus 2 reduces the number of adverbial chunks comparatively to corpus 1 and induces a significant improvement of F-measure for this chunk: AdP has a F-measure of 58,14 in the first experiment, 71,87 in the second one and 95,76 in the third one! When the POS I is correct, the IntP is very well identified: its F-measure is 99,4.

The remaining errors concern either repetitions which are not correctly treated or interjections inside nominal chunks, like the example of section 3.4 wrongly chunked as: (l'/DET école/NC)$_{NP}$ (euh/I)$_{I}$ (publique/ADJ)$_{AP}$ (see note 10).

More generally, we see that an excellent POS tagger learnt from written sentences makes about 20% errors on transcribed data, and induces the same level of errors for the chunker. Correcting the POS errors helps the chunker by improving its score of about 10%. But learning a specific chunker from corrections at both levels is even better, even with limited training data.

References

[Abn91] Abney, S.: Parsing by chunks. In: Berwick, R., Abney, R., Tenny, C. (eds.) Principle-based Parsing. Kluwer Academic Publisher (1991)

[ACT03] Abeillé, A., Clément, L., Toussenel, F.: Building a treebank for french. In: Abeillé, A. (ed.) Treebanks. Kluwer, Dordrecht (2003)

[AMF08] Antoine, J.-Y., Mokrane, A., Friburger, N.: Automatic rich annotation of large corpus of conversational transcribed speech: the chunking task of the epac project. In: Proceedings of LREC 2008 (May 2008)

[BB05] Blanche-Benveniste, C.: Sémantique de l'oral, chapter Sémantique et corpus. Les aspects dynamiques de la composition sémantique de l'oral (2005)

[BBJ87] Blanche-Benveniste, C., Jeanjean, C.: Le français parlé, transcription et édition. Didier Erudition (1987)

[BCDW10] Blanc, O., Constant, M., Dister, A., Watrin, P.: Partial parsing of spontaneous spoken french. In: Proceedings of LREC 2010 (2010)

[CC08] Crabbé, B., Candito, M.H.: Expériences d'analyse syntaxique statistique du français. In: Actes de TALN 2008 (2008)

[CT12] Constant, M., Tellier, I.: Evaluating the impact of external lexical resources unto a crf-based multiword segmenter and part-of-speech tagger. In: Proceedings of LREC 2012 (2012)

[LCY10] Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: Proceedings of ACL 2010, pp. 504–513. Association for Computational Linguistics (July 2010)

[LMP01] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML 2001, pp. 282–289 (2001)

[SP03] Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proceedings of HLT-NAACL, pp. 213–220 (2003)

[TETP10] Tellier, I., Eshkol, I., Taalab, S., Prost, J.P.: Pos-tagging for oral texts with crf and category decomposition. Research in Computing Science 46, 79–90 (2010)

[VV99] Valli, A., Veronis, J.: Etiquetage grammatical des corpus de parole: problèmes et perspectives. Revue Française de Linguistique Appliquée 4(2), 113–133 (1999)

SVD Based Graph Regularized Matrix Factorization*

Ephrime A. Vidar[1] and Sweedy K. Alvindia[1,2]

[1] Department of Mathematics and Computer Science
University of the Philippines Baguio, Baguio City 2600, Philippines
[2] State Key Laboratory for Novel Software Technology
Nanjing University, P.R. China
sweedyk.alvindia@yahoo.com.ph

Abstract. Graph regularized matrix factorization has been proposed by Cai et al. (2011) with an alternate optimization strategy due to the difficulty of close form solution. In this paper, we develop a novel method for graph regularized matrix factorization. This method is based on Singular Value Decomposition (SVD), and its solution is close formed. We carried out experiments on a database for the task of clustering problem to shown the advantage of the proposed method.

Keywords: Matrix Factorization, Graph Regularization, Singular Value Decomposition.

1 Introduction

Matrix factorization is a technology for data representation, which is very important to the problem of pattern recognitions and information retrieval [1–4]. The factorization of data matrix can show the latent structure of the data set [5]. Given a data matrix X, and a a rank R for the factorization, a matrix factorization problem is usually formulated using a Frobenius norm [6, 7],

$$\begin{aligned} &min_{H,W} \ \|X - HW\|_F^2 \\ &subject\ to\ H^\top H = I_r. \end{aligned} \tag{1}$$

where I_r is the $r \times r$ identity matrix. This problem can solved as a Principle Component Analysis (PCA) problem [8–10], or a Singular Value Decomposition (SVD) problem [11–13]. The problem of matrix factorization is shown in Fig. 1.

Nonnegative matrix factorization (NMF) [14–17] is is a case of matrix factorization, which is developed for nonnegative data. Its formula is summarized as follows,

$$\begin{aligned} &min_{H,W} \ \|X - HW\|_F^2 \\ &subject\ to\ H \geq 0, W \geq 0. \end{aligned} \tag{2}$$

* This work was supported by the State Key Laboratory for Novel Software Technology (Grant No. KFKT2012B17).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 234–241, 2013.

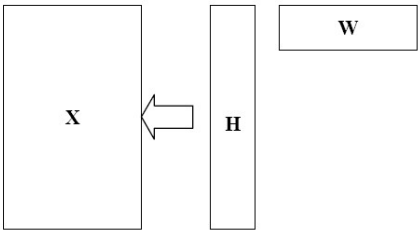


Fig. 1. Matrix Factorization Problem

It seeks a new representation for the data to make the basis matrix H and the coefficient matrix W to be nonnegative. In this way, some physical interpretation can be obtained.

The possible inherent manifold structure of the data has been ignored by the traditional matrix factorization methods [18–20]. The low dimensional representation V will loss thus manifold information. Thus we use the graph regularization to solve this problem [21–23].,

$$\begin{aligned} min_W \ & \sum_{i,j} \|w_i - w_j\|_F^2 A_{ij} \\ & = Tr(WBW^\top) \end{aligned} \tag{3}$$

where B is the Laplacian matrix of the graph [24–26], and A is the graph weight matrix. It has been used in Locality Preserving Projection (LPP) [27–30].

Recently, the graph regularized matrix factorization has been proposed by Cai et al. [31] so that the graph structure of the data can be considered when the new representation is learning. They combine the original loss function and a graph regularization term for the matrix factorization problem. In this way, the new representation of the data can keep the graph structure somehow. This method has been improved to various versions. For example, in [32], Wang et al. learned the optimal graph by combining several candidate graphs. In [33], Du et al. used the weighted graph regularized nonnegative matrix factorization for problem of luster ensembles. In [34], Wang et al. proposed to use the feature selection method to refine the graph to regularized the matrix factorization. In [35], Rajabi et al. developed a new method for hyperspectral data unmixing problem by using graph regularized nonnegative matrix factorization.

However, the solution of this problem is difficult and usually an iterative algorithm is adopted to solve it. In this paper, we proposed a novel method which can provide a close form solution.

The rest part of this paper is organized as follows. In Section 2, we introduce the problem and its closed form solution. The experimental results are given in Section 3. Some conclusions is given in Section 4.

2 Graph Regularized Matrix Factorization

We assume that $X \in R^{d \times n}$ is a data matrix with its columns are data vectors. Moreover, $B \in R^{n \times n}$ is a regularization matrix to implement the graph regularization. The problem of graph regularized matrix factorization is formulated as follows:

$$\begin{aligned} &min_{H,W} \ \|X - HW\|_F^2 + \alpha Tr(WBW^\top) \\ &subject\ to\ H^\top H = I_r. \end{aligned} \tag{4}$$

where $H \in R^{d \times r}$ is an orthonormal matrix contains the orthogonal basis vectors, and $V \in R^{r \times n}$ is the representation matrix with each column as a new representation. We introduce a new variable $U = HW$ to replace W so that the problem could be easier. Thus the objective function in (4) can be rewritten as,

$$\begin{aligned} &\|X - U\|_F^2 + \alpha Tr(WBW^\top) \\ &= \|X - U\|_F^2 + \alpha Tr(H^\top HWBW^\top) \\ &= \|X - U\|_F^2 + \alpha Tr(HWBW^\top H^\top) \\ &= \|X - U\|_F^2 + \alpha Tr(UBU^\top) \end{aligned} \tag{5}$$

by substituting it back to (4), we have

$$\begin{aligned} &min_U \ \|X - U\|_F^2 + \alpha Tr(UBU^\top) \\ &subject\ to\ rank(U) \leq r. \end{aligned} \tag{6}$$

We further rewrite the objective in (6) as

$$\begin{aligned} &\|X - U\|_F^2 + \alpha Tr(UBU^\top) \\ &= Tr[(X - U)(X - U)^\top] + \alpha Tr(UBU^\top) \\ &= Tr(XX^\top) - 2Tr(XU^\top) + Tr(UU^\top) + \alpha Tr(UBU^\top) \\ &= Tr(XX^\top) - 2Tr(XU^\top) + Tr[U(I + \alpha B)U^\top] \end{aligned} \tag{7}$$

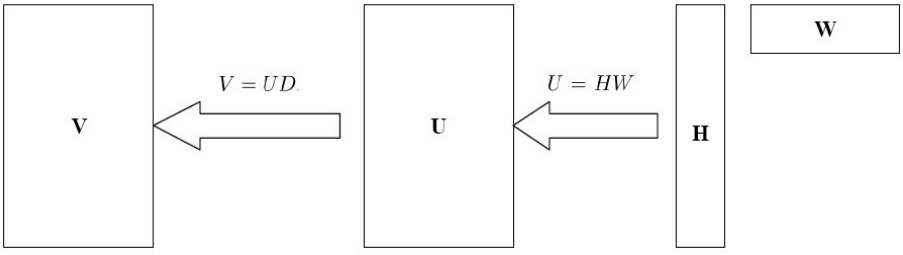


Fig. 2. Transforming the solution from W to V

We define $C = (I + \alpha B)$, and $DD^\top = C$ then,

$$\begin{aligned} & Tr(XX^\top) - 2Tr(XU^\top) + Tr[U(I + \alpha B)U^\top] \\ & = Tr(XX^\top) - 2Tr(XU^\top) + Tr[UCU^\top] \\ & = Tr(XX^\top) - 2Tr(XU^\top) + Tr[UDD^\top U^\top] \\ & = Tr(XX^\top) - 2Tr[(XD^{\top -1})(UD)^\top] + Tr[(UD)(UD)^\top] \end{aligned} \tag{8}$$

we also define $V = UD$, it turns to

$$\begin{aligned} & Tr(XX^\top) - 2Tr[(XD^{\top -1})(UD)^\top] + Tr[(UD)(UD)^\top] \\ & = Tr(XX^\top) - 2Tr[(XD^{\top -1})V^\top] + Tr[VV^\top] \\ & = Tr(XX^\top) - 2Tr[(XD^{\top -1})V^\top] + Tr[VV^\top] \\ & = Tr(XX^\top) - Tr[(XD^{\top -1})(XD^{\top -1})^\top] \\ & \quad + Tr[(XD^{\top -1})(XD^{\top -1})^\top] - 2Tr[(XD^{\top -1})V^\top] + Tr[VV^\top] \\ & = Tr(XX^\top) - Tr[(XD^{\top -1})(XD^{\top -1})^\top] \\ & \quad + \left\| (XD^{\top -1}) - V \right\|_F^2 \end{aligned} \tag{9}$$

Substituting it to (6), and removing the terms irrelevant to V, we have

$$\begin{aligned} & min_V \ \left\| (XD^{\top -1}) - V \right\|_F^2 \\ & subject\ to\ rank(V) \leq r. \end{aligned} \tag{10}$$

This procedure is shown in Fig. 2. Then we perform SVD to $(XD^{\top -1})$,

$$(XD^{\top -1}) = E\Sigma F^\top \tag{11}$$

To approximate V with $rank(V) \leq r$, we have

$$V^* = \widetilde{E}\widetilde{\Sigma}F^\top \tag{12}$$

where $\widetilde{\Sigma}$ contains only the r largest singular values, and $\widetilde{E}$ contains only the singular vectors corresponding to the r largest singular values. So we have

$$V^* \approx (XD^{\top -1}) \tag{13}$$

Moreover, because $\widetilde{E}\widetilde{E}^\top = I$,

$$\begin{aligned} V^* &\approx \widetilde{E}\widetilde{E}^\top (XD^{\top -1}) \\ &\Rightarrow U^* D \approx \widetilde{E}\widetilde{E}^\top (XD^{\top -1}) \\ &\Rightarrow U^* \approx \widetilde{E}\widetilde{E}^\top (XD^{\top -1}) D^{-1} \\ &\Rightarrow U^* \approx \widetilde{E}\widetilde{E}^\top XC^{-1} \\ &\Rightarrow H^* W^* \approx \widetilde{E}\widetilde{E}^\top XC^{-1} \end{aligned} \tag{14}$$

Thus we set

$$\begin{aligned} H^* &= \widetilde{E} \\ W^* &= \widetilde{E}^\top XC^{-1} \end{aligned} \tag{15}$$

Which is the closed form solutions.

3 Experiments

We evaluated our algorithm on clustering problem. The experiments are conducted on an image database, which is the COIL20 dataset [36]. It is composed of images of 20 different objects, and there are 72 images for each object in this database. Each image is of size of 32×32 pixels. Some images from this dataset is shown in Fig. 3. We use this database because it has a manifold structure clearly.

Using this dataset, we compare our algorithm to the following ones:

1. Matrix factorization using SVD [37, 38],
2. Nonnegative Matrix Factorization (NMF) [1] and
3. Graph regularized NMF (GNMF) [31].

The clustering accuracies are given in Fig. 4. It can be seen that proposed method obtained the best clustering performances all the time. Moreover, it's also interesting to see that GNMF outperforms other methods.

Fig. 3. Images of COIL20 database

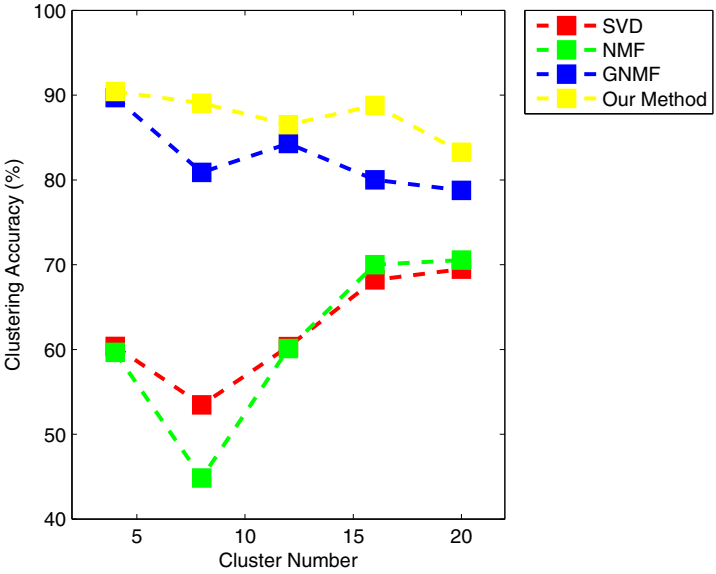


Fig. 4. Clustering results on the database using different methods

4 Conclusion

In this paper, we presented a novel graph regularized matrix factorization method. It used the SVD to obtained a closed form solution. The experiment showed its advantage over other methods. In fact it can also be extended into sparse matrix factorization methods, by using so called K-SVD methods [39–41].

References

1. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
2. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5(2), 111–126 (1994)
3. Brunet, J.P., Tamayo, P., Golub, T., Mesirov, J.: Metagenes and molecular pattern discovery using matrix factorization. Proceedings of the National Academy of Sciences of the United States of America 101(12), 4164–4169 (2004)
4. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. SIGIR Forum (ACM Special Interest Group on Information Retrieval) (SPEC. ISS.), 267–273 (2003)
5. Bishop, C., Tipping, M.: A hierarchical latent variable model for data visualization. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 281–293 (1998)

6. Carpentieri, B., Duff, I., Giraud, L.: Sparse pattern selection strategies for robust frobenius-norm minimization preconditioners in electromagnetism. Numerical Linear Algebra with Applications 7(7-8), 667–685 (2000)
7. Ma, C., Kamp, Y., Willems, L.F.: Frobenius norm approach to glottal closure detection from the speech signal. IEEE Transactions on Speech and Audio Processing 2(2), 258–265 (1994)
8. Fung, C.P., Kang, P.C.: Multi-response optimization in friction properties of pbt composites using taguchi method and principle component analysis. Journal of Materials Processing Technology 170(3), 602–610 (2005)
9. Zhao, S., Zhang, J., Xu, Y.: Monitoring of processes with multiple operating modes through multiple principle component analysis models. Industrial and Engineering Chemistry Research 43(22), 7025–7035 (2004)
10. Chen, S., Zhu, Y.: Subpattern-based principle component analysis. Pattern Recognition 37(5), 1081–1083 (2004)
11. Edfors, O., Sandell, M., Van Beek, J.J., Wilson, S., Brjesson, P.: Ofdm channel estimation by singular value decomposition. IEEE Transactions on Communications 46(7), 931–939 (1998)
12. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications 21(4), 1253–1278 (2000)
13. Alter, O., Brown, P., Botstein, D.: Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences of the United States of America 97(18), 10101–10106 (2000)
14. Virtanen, T.: Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. IEEE Transactions on Audio, Speech and Language Processing 15(3), 1066–1074 (2007)
15. Wang, J.J.Y., Wang, X., Gao, X.: Non-negative matrix factorization by maximizing correntropy for cancer clustering. BMC Bioinformatics 14 (2013)
16. Berry, M., Browne, M., Langville, A., Pauca, V., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis 52(1), 155–173 (2007)
17. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Computation 19(10), 2756–2779 (2007)
18. Vilamala, A., Lisboa, P., Ortega-Martorell, S., Vellido, A.: Discriminant convex non-negative matrix factorization for the classification of human brain tumours. Pattern Recognition Letters (2013)
19. Li, Z., Liu, J., Lu, H.: Structure preserving non-negative matrix factorization for dimensionality reduction. Computer Vision and Image Understanding (2013)
20. Hu, L., Wu, J., Wang, L.: Linear projective non-negative matrix factorization. ICIC Express Letters 7(8), 2285–2291 (2013)
21. Lezoray, O., Elmoataz, A., Bougleux, S.: Graph regularization for color image processing. Computer Vision and Image Understanding 107(1-2), 38–55 (2007)
22. Weinberger, K., Sha, F., Zhu, Q., Saul, L.: Graph laplacian regularization for large-scale semidefinite programming. In: Advances in Neural Information Processing Systems, pp. 1489–1496 (2007)
23. Zhang, T., Popescul, A., Dom, B.: Linear prediction models with graph regularization for web-page categorization. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 2006, pp. 821–826 (2006)
24. Li, J.S., Zhang, X.D.: A new upper bound for eigenvalues of the laplacian matrix of a graph. Linear Algebra and its Applications 265(1-3), 93–100 (1997)

25. Trinajstić, N., Babić, D., Nikolić, S., Plavšić, D., Amić, D., Mihalić, Z.: The laplacian matrix in chemistry. Journal of Chemical Information and Computer Sciences 34(2), 368–376 (1994)
26. Merris, R.: Laplacian matrices of graphs: a survey. Linear Algebra and its Applications 197-198(C), 143–176 (1994)
27. Li, J.B., Pan, J.S., Chu, S.C.: Kernel class-wise locality preserving projection. Information Sciences 178(7), 1825–1835 (2008)
28. Chen, S., Zhao, H., Kong, M., Luo, B.: 2D-lpp: A two-dimensional extension of locality preserving projections. Neuro Computing 70(4-6), 912–921 (2007)
29. Yu, W., Teng, X., Liu, C.: Face recognition using discriminant locality preserving projections. Image and Vision Computing 24(3), 239–248 (2006)
30. Hu, D., Feng, G., Zhou, Z.: Two-dimensional locality preserving projections (2dlpp) with its application to palmprint recognition. Pattern Recognition 40(1), 339–342 (2007)
31. Cai, D., He, X., Han, J., Huang, T.: Graph regularized non negative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1548–1560 (2011)
32. Wang, J.J.Y., Bensmail, H., Gao, X.: Multiple graph regularized nonnegative matrix factorization. Pattern Recognition 46(10), 2840–2847 (2013)
33. Du, L., Li, X., Shen, Y.-D.: Cluster ensembles via weighted graph regularized nonnegative matrix factorization. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS (LNAI), vol. 7120, pp. 215–228. Springer, Heidelberg (2011)
34. Wang, J.Y., Almasri, I., Gao, X.: Adaptive graph regularized nonnegative matrix factorization via feature selection. In: Proceedings of the International Conference on Pattern Recognition, pp. 963–966 (2012)
35. Rajabi, R., Khodadadzadeh, M., Ghassemian, H.: Graph regularized nonnegative matrix factorization for hyperspectral data unmixing. In: Proceedings of the 2011 7th Iranian Conference on Machine Vision and Image Processing, MVIP 2011 (2011)
36. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96 (1996)
37. Boutsidis, C., Gallopoulos, E.: Svd based initialization: A head start for nonnegative matrix factorization. Pattern Recognition 41(4), 1350–1362 (2008)
38. Hong, Z., Zhen, L., Zhao, L.: Non-negative matrix factorization based on double sparsity k-svd. Applied Mechanics and Materials 190-191, 352–355 (2012)
39. Aharon, M., Elad, M., Bruckstein, A.: K-SVD and its non-negative variant for dictionary design. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 5914, pp. 1–13 (2005)
40. Bryt, O., Elad, M.: Compression of facial images using the k-svd algorithm. Journal of Visual Communication and Image Representation 19(4), 270–282 (2008)
41. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)

Clustering, Noise Reduction and Visualization Using Features Extracted from the Self-Organizing Map

Leonardo Enzo Brito da Silva and José Alfredo Ferreira Costa

Universidade Federal do Rio Grande do Norte
Programa de Pós-Graduação em Engenharia Elétrica e de Computação
Natal, RN, Brasil
{leonardoenzob,jafcosta}@gmail.com

Abstract. This paper presents an analysis of a feature space generated by extracting properties related to pattern density and Euclidean distances between neurons from the self-organizing map network. Hence, along with the weight vector, each neuron has a 2-D feature vector associated with it, whose components are extracted from the U-matrix and a hit matrix, where latter is based on hyperspheres centered on each neuron. This collection of feature vectors, that represents the neurons of the network, is partitioned into different groups, and their labels are carried back to the data space as well as the neuron grid, in order to perform the tasks of clustering, noise reduction and visualization. Experiments were carried out using synthetic and real world data sets.

Keywords: self-organizing maps, clustering, noise reduction, visualization, feature extraction.

1 Introduction

The increasingly quantity of data produced in the modern world have maximized the necessity for understanding and exploiting the information existent [1][2], while simultaneously overcoming its quality related problems. The self-organizing maps (SOM) [3][4] are artificial neural networks widely used in the data mining field for partitioning the data into similar groups and as a tool for visualization through low dimensionality projections of multidimensional data, due to the fact that classic clustering algorithms may be applied to the SOM neurons [5], and many visualization techniques associates data characteristics to the topologically ordered neuron grid, so that an insight of the data distribution may be obtained [6].

This paper focuses on performing the clustering task, noise filtering and visualization through partitioning the self-organizing map according to the distribution of feature vectors associated with the neurons, which enclose characteristics related to pattern density and distances between neurons.

The paper is organized as follows. Section 2 provides a brief review of the SOM network and related visualization methods; also the derived feature space is defined. In Section 3, the proposed approach is described. The results of the experiments and discussions are presented in Sections 4 and 5, respectively.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 242–251, 2013.

2 Self-Organizing Maps and Derived Feature Space

The self-organizing maps are neural networks based on unsupervised learning. They are composed by a lattice of neurons, each one associated with a weight vector $\boldsymbol{w_i}$ in the p-dimensional data space. In this work, the SOM is trained using the *batch mode,* in which the whole data set is fed at once to the network. At each epoch, the BMUs (neurons with the smallest Euclidean distance from their associated weight vectors to the input patterns) for all patterns are determined, so the M neurons of the network can be simultaneously updated according to (1):

$$\boldsymbol{w}_j(t+1) = \frac{\sum_{i=1}^{N} h_{j,c}(t)\mathbf{x}_i}{\sum_{i=1}^{N} h_{j,c}(t)} \tag{1}$$

where t denotes the iteration, $\boldsymbol{w}_j(t)$ is a weight vector associated with the j[th] neuron, $\mathbf{x}_i$ is the i[th] input pattern, N is the total number of patterns, and $h_{j,c}(t)$ is the neighborhood kernel, which is usually a Gaussian function defined by the neighborhood radius σ.

In order to inspect the relative sizes and positions of clusters in a given data set, visualization techniques must be applied to a trained SOM network, which are typically matrix plots of Euclidean distances between neurons or pattern density, such as the popular U-matrix [7] and P-matrix [8], respectively. The U-matrix consists of an image that portrays the Euclidean distances in the data space between the SOM neurons. Regions of small and large distances are often regarded as clusters and their borders, respectively. The P-matrix is generated by counting the number of patterns inside a Pareto hypersphere centered on each neuron.

The proposed feature space consists of the following concept: for each neuron in the grid, besides the weight vector in the data space, there is also associated with it a 2-D feature vector, which carries meaningful information from the data set. The characteristic of the feature vector distribution is analyzed so as to diminish the noise present in the data set, to perform a clustering task or to visualize similar groups of neurons. In this work, the feature vector $\boldsymbol{f_i}$ of a neuron $\boldsymbol{w_i}$ has two components:

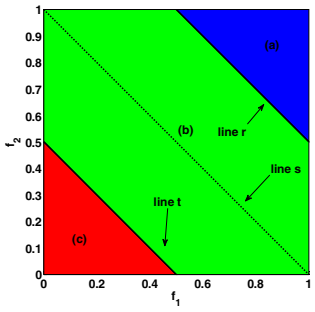
$$\boldsymbol{f_i} = [f_1 \quad f_2] \tag{2}$$

The first component (f_1) is the value extracted from the neuron's corresponding position in U-matrix. However, it is negated with the purpose of becoming a measure of similarity instead of dissimilarity. The second component (f_2) is obtained from the matrix plot in which every position entails the number of patterns inside a hypersphere centered in that neuron, where the radius is determined as the minimum so all neurons have at least one pattern inside its respective hypersphere. This approach is inspired from the P-matrix visualization method. The radius of this hit matrix H is then defined as:

$$r = \max_j \left(\min_i \left\| \mathbf{x}_i - \boldsymbol{w}_j \right\| \right) \tag{3}$$

where $i = (1,2,\ldots,N)$, and $j = (1,2,\ldots,M)$.

It is assumed that, in general, the feature space should comprise three main regions (Fig. 1), which correspond to neurons that are positioned in the core of the clusters (region 1), in their frontier (region 2) or between them (region 3). The values of the components are normalized in the range $[0;1]$.



Components		f_1	
		High	*Low*
f_2	*High*	Cluster's neurons (a)	Boundary neurons (b)
	Low	Boundary neurons (b)	Interpolating neurons (c)

Fig. 1. The three colored regions correspond to regions that enclose interpolating (region 3 - red), boundary (region 2 - green) and cluster's neurons (region 1 - blue). The lines *r*, *s*, and *t* symbolize generic hyperplanes that divide the feature space.

3 Proposed Approach

After training a SOM network, the feature vectors are generated with the values obtained from the U-matrix and the hit matrix H, so an analysis can be carried out to perform noise reduction, clustering or visualization. The main goal is to identify interpolating neurons, i.e. those neurons that do not belong to a cluster and just link close-knit groups of neurons, which represent the core of the clusters.

In order to partition the data of the feature space, the k-means algorithm [9] and a competitive network [10] were used. The assumption described in Section 2 was simplified in the sense that only two regions were considered, that is, the generic line *s* of Fig. 1, so one part of the boundary neurons would be included in the interpolating neurons subset and the other to the clusters' neurons subset. Although not critical to most of the data sets used in the experiments, this simplification of the hypothesis was not functional considering two data sets, and thus it cannot be generally used. If the points in the feature space regarding data sets with noise may be modeled as an exponential function with a reasonable goodness-of-fit statistics of the curve fitting, then the partitioning of the feature vectors into two subsets corresponding to the interpolating neurons and the clusters' neurons may also be accomplished using the common elbow criterion as a threshold. After the segmentation in the feature space, the subset with the largest median of the Euclidean norm is considered as the core of the clusters. For the purpose of reducing the noise, the final step consists of determining the

patterns from which each neuron is related, and then eliminate from the data set the ones that are associated with interpolating neurons. Conversely, when the task being considered is clustering, then the clusters are identified and labeled in the SOM grid using 4-neigborhood connected components labeling (CCL) [11]. Regions that are not connected and are smaller than a percentage of the total number of SOM neurons (α) are disregarded. All the remaining neurons are classified through the k Nearest Neighbor algorithm (kNN) [12], in which the parameter k was set to 1. At last, the patterns are labeled using the partitioned SOM.

In the case that a data set has several close clusters with extremely different densities, the proposed method may not be directly applied as described. Nonetheless, different regions of similar neurons of the SOM may be highlighted by visualizing the partitioning of the feature space into different number of regions.

4 Experiments

The SOM Toolbox [13] was used to implement the SOM networks, which were trained using the batch mode (1000 epochs) and linear initialization. The Table 1 sums up the characteristics of the data sets used in the experiments, which are depicted in Fig. 2. They come from [14-17], and the data set *D2* was artificially generated. Such databases were preprocessed using the linear normalization, i.e. the data sets' attributes were normalized in the $[0;1]^n$ cube.

Table 1. Data sets characteristics

Data set	*Dim.*	*Size*	*Clusters*	*Type*	*Main problem*
DS3	2	10000	9	Synthetic	Clusters with different shapes / noise
Engytime	2	4096	2	Synthetic	Overlapping clusters
Hepta	3	212	7	Synthetic	Clusters with different variances
Tetra	3	400	4	Synthetic	Very close clusters
Twodiamonds	2	800	2	Synthetic	Clusters connected by a bridge
Wingnut	2	1016	2	Synthetic	Clusters with variable densities
D2	4	600	4	Synthetic	Data set with high dimensionality
D3	2	1500	5	Synthetic	Gaussian clusters
Wine	13	178	3	Real World	Data set with high dimensionality

The analysis carried out over the feature space using the proposed approach is presented in detail considering the *DS3* data set through the Figs. 3 to 7. A SOM network of size 100x100 was trained with the parameters previously described and final neighborhood radius equal to zero. The Fig. 3 depicts the U-matrix and P-matrix generated using the SOMVIS Package, as well as the proposed hit matrix H. Considering the Fig. 3 'b' and 'c', both of which contains information of pattern density, the definition of the clusters boundaries are sharper in item 'c', and thus the feature space is generated using the values of the matrixes of the items 'a' and 'c' (Fig. 4a).

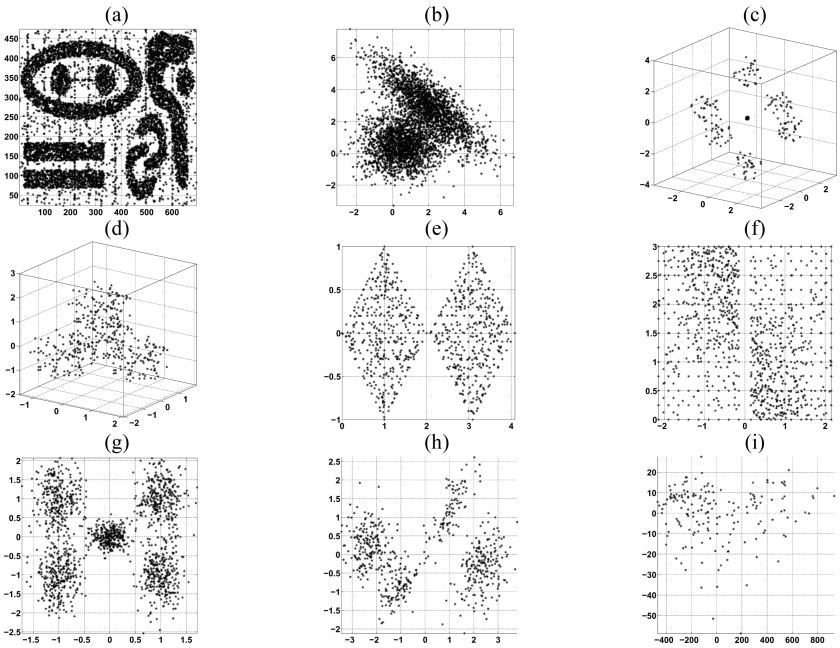


Fig. 2. Illustration of the data sets: (a) *DS3*, (b) *Engytime*, (c) *Hepta*, (d) *Tetra*, (e) *Twodiamonds*, (f) *Wingnut*, (g) *D3*, (h) *D2* and (i) *Wine*, both with a 2-D PCA projection.

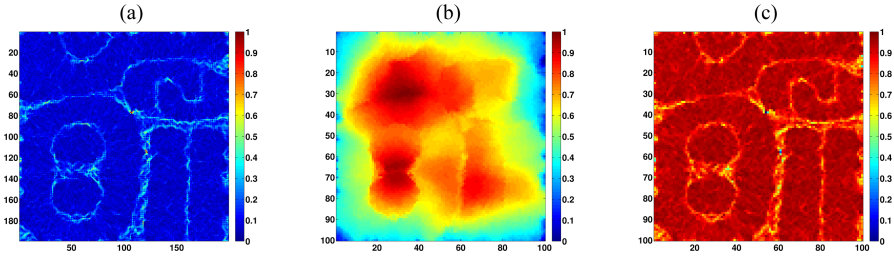


Fig. 3. (a) U-matrix, (b) P-matrix, and (c) H-matrix (with the radius defined by Eq. 3) of the SOM network trained with the *DS3* data set.

The next stage consists of partitioning the feature space into two regions of similar neurons. The region with the largest median of the Euclidean norm is regarded as the one representing the core of the data set's clusters. This may be achieved using any kind of clustering algorithm; in the case of this particular map, the k-means algorithm was applied. If the objective is to perform the noise filtering, then patterns associated with the interpolating neurons (mostly noise) are excluded from the data set (Fig. 4c).

The feature space may reveal different regions of neurons with similar characteristics as a function of the number k of the partitions of this space, and therefore it can be used as a tool for visualization. In Fig. 5, the dependence of the method with the value of the parameter k is depicted. As expected, when partitioning the feature space into 2 regions, the boundary neurons are included in both the interpolating neurons

subset and core of the clusters subset. Considering k equal to 3 leads to the appearance of the boundary neurons as a group of its own. Finally, increasing the value of k turns stricter the neurons' intra-group similarity.

In order to automatically find the clusters of the *DS3* data set, the CCL was applied to the image of Fig. 5a, and the result obtained is depicted in Fig. 6a. After labeling the group of neurons related to the clusters, the interpolating neurons may be labeled by flooding or by the *k*NN algorithm, and in this work the latter is used (Fig. 7a). At last, the segmented SOM is used to label the data set.

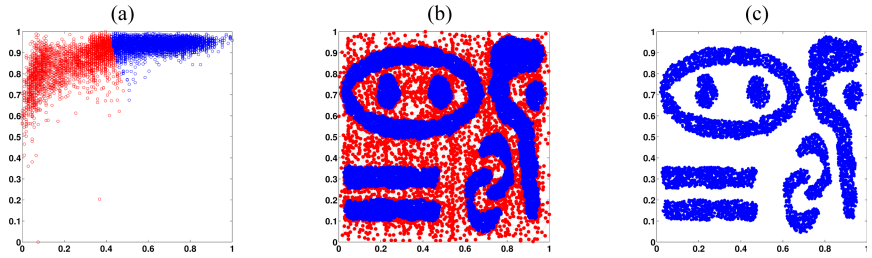


Fig. 4.(a) Feature space data divided into 2 regions using the k-means algorithm. (b) The 2 groups of neurons viewed in the data space. Classified neurons of the SOM network are shown in red (interpolating neurons subset) and blue (cluster's neurons subset). Neurons of the same group are represented with the same colors in items 'a' and 'b'. (c) Filtered data set.

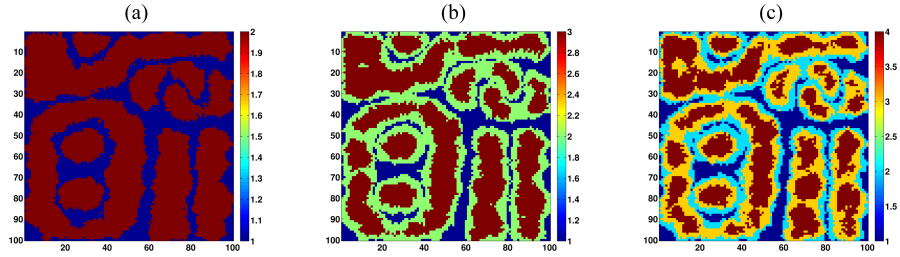


Fig. 5. Matrix plots with the same size as the network lattice. The partitions of the feature space are shown for (a) k = 2, (b) k = 3, and (c) k = 4, in which each color correspond to neurons of the same group.

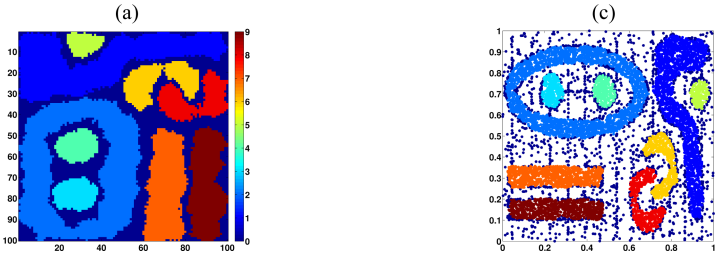


Fig. 6. (a) Labeled neurons of the SOM grid using CCL. (b) Labeled data set.

The Fig. 7 also depicts the results of partitions obtained by the watershed [18] and k-means algorithms, the latter applied over the SOM neurons in the data space, with the parameter k defined by the best value of the Davies-Bouldin index (DBI) [19]. The watershed algorithm was applied to the U-matrix image of Fig. 3a after a morphological image processing (filtering through area open and area close) [20], where the area size was set to half the maximum dimension of the map [17]. The proposed method was able to uncover the 9 clusters of the *DS3* data set, whereas the watershed and k-means algorithms found 8 and 30 clusters, respectively.

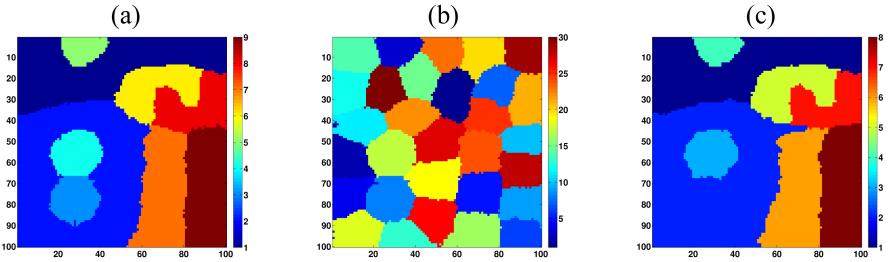


Fig. 7. (a) Labeled SOM neurons trough the (a) k-means in the feature space with posterior CCL, (b) k-means in the data space over the SOM neurons, with the DBI criterion for the choice of the parameter k, (c) watershed algorithm over the U-matrix of Fig. 3a.

In order to perform the clustering task for the remaining data sets, several experiments were carried out using diverse values for the following parameters: map size, grid type, final neighborhood radius (σ_f), the minimum cluster size (α), and number of neuron regions (k). The majority of the map sizes was defined according to [3][5] and ended up to be rectangular, the others were defined as square maps, as they led to better results (Table 2). Different sizes were considered as multiples of the original size. The scale factor S is such that $S = \{0.75,1,1.25,1.5,1.75,2\}$.

Table 2. Summary of the parameters used throughout the experiments

Data set	*Wine*	*Engytime*	*Hepta*	*Tetra*	*Twodiamonds*	*Wingnut*	*D2*	*D3*
Map size (S=1)	8x8	21x15	10x10	10x10	20x7	10x10	18x7	16x12
k	2	2	3	3	2	2	2	2
α (%)	1	1	1	5	1	5	1	1
$*\sigma_f$	1	1	1	1	0	1	1	1

*Final neighborhood radius during the SOM training.

The k-means and competitive network were chosen to perform the partition of the representation of the neurons in the feature space. Due to the random initialization of the k-means algorithm, it was repeated 100 times and the result with the smallest sum of squared errors was selected. The competitive network was trained with 100 epochs. The results obtained were compared to the same methods applied to the SOM neurons in the data space, using the same parameter settings. In this case, the parameter k ranged from 2 to $\sqrt{M}$, where M is the total number of neurons of the network, and again, the one with best DBI value was chosen. The Watershed algorithm applied to

the U-matrix image of each trained SOM. The Fig. 8 depicts the results obtained while varying the mentioned parameters. The classification accuracy (CA) [21] was used for evaluating the results (Fig. 8). The CA consists of the percentage of the properly classified patterns with respect to the complete data set.

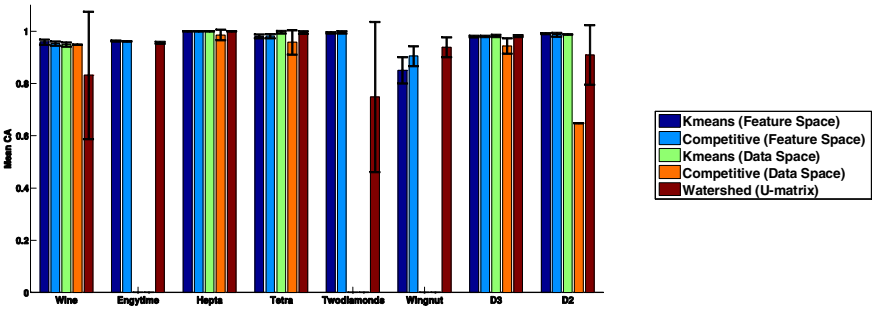


Fig. 8. Mean classification accuracy and standard deviation, for the clustering task performed by the k-means and competitive network both in the data set space and the feature space, as well as the watershed algorithm applied to the U-matrix generated by the trained SOMs.

The k-means and competitive networks applied to the SOM neurons in the data space along with the DBI were not able to find the correct number of clusters in none of the 6 maps (scale factor S) for the data sets *Engytime*, *Twodiamonds* and *Wingnut*. The difficulties encountered by the watershed algorithm for the *Twodiamonds* data set rely in the fact that this data set has connected clusters. Thus, in this case density metrics are more relevant than distance metrics, which is the one displayed in the U-matrix that is used for segmentation. The watershed algorithm was used after the same previously described morphological image processing. It must be noted that improved results using this algorithm may be obtained by thoroughly fine tuning the area size used in the filtering stage.

As several maps were trained, additional measures related to the clustering task were made necessary: the correct clustering frequency (CF) and clustering efficiency (CE). The CF measures the percentage of runs in which the correct number of clusters was found, because the CA is only calculated if and only if the correct number of clusters was found. The CE gives an overall performance measure of the clustering ($CE = Mean\ CA \times CF$), as it takes into account the number of times the right number of cluster was found and how correct was the classification. The CFs obtained are depicted in detail in Table 4 for all the clustering methods used in the experiments.

Although the methods have shown a good overall performance in terms of mean classification accuracy, it must be noted that clustering the feature space and going back to apply the results in the data space demonstrated itself more consistent due to the fact that the correct number of clusters was identified in more simulations than the other methods (see the CE depicted in the radar plot of Fig. 9). For instance, the lowest CFs considering the k-means and the competitive network used for clustering the feature space are 67% and 83% of the runs, respectively, whereas the lowest percentage is 0% for the same algorithms applied to the neurons of the SOM in the data space, or 33% when using watershed. Besides, the mean classification accuracy was above 0.8 for all tested data sets, with comparatively small standard deviations.

Table 3. Classification Frequency

Data set	*Feature Space*		*Data Set Space*		
	k-means	Competitive	k-means	Competitive	Watershed
Wine	1.00	1.00	0.33	0.17	0.67
Engytime	1.00	0.83	0.00	0.00	1.00
Hepta	0.83	0.83	0.67	0.67	1.00
Tetra	1.00	1.00	1.00	0.67	0.83
Twodiamonds	0.67	1.00	0.00	0.00	0.67
Wingnut	1.00	1.00	0.00	0.00	0.83
D3	1.00	1.00	0.83	0.83	1.00
D2	1.00	0.83	0.33	0.17	0.33

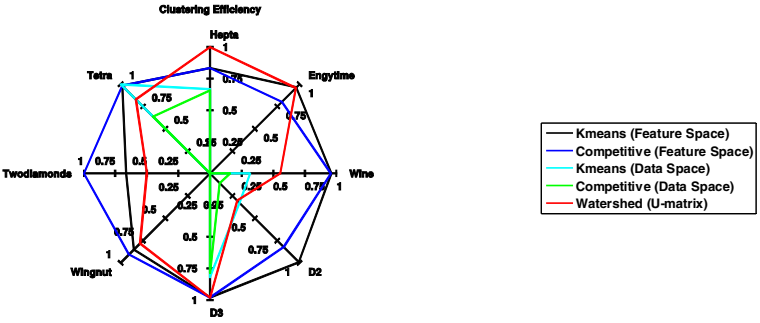


Fig. 9. Classification efficiency for the clustering task performed by the k-means and competitive network both in the data set space and the feature space, as well as the watershed algorithm applied to the U-matrix generated by the trained SOMs.

5 Conclusions

An analysis of the feature space generated by extracting properties related to density and distances between neurons of the SOM network was presented. Applications of the feature space include data filtering so as to reduce noise, as well automatically detecting the number of clusters and used as a tool for visualization.

In order to partition the data set or to reduce the noise, classical clustering algorithms may be used as well as the elbow method if an exponential curve fit is feasible. In this work the k-means algorithm and the competitive network were used with the simplified assumption that in most data sets exist only two types of neurons, however this hypothesis may not be considered universal, as it worked with the great majority, but not all data sets, where the separation in three regions is necessary. The results were compared to the same clustering algorithms applied to the neurons of the SOM network in the data space, as well as with the watershed technique. The proposed method was able to consistently segment the SOM into the correct number of clusters with high classification accuracies. If the clusters present in a given data set have utterly different variances, then the method may not be applied. Nevertheless, neurons with similar characteristics may be depicted as a matrix plot for a given number of partitions of the feature space (k), and therefore provide clues to the data clusters' relative positions and sizes.

References

1. Larose, D.T.: Discovering Knowledge in Data. John Wiley & Sons (2005)
2. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2006)
3. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2001)
4. Kohonen, T.: Essentials of the self-organizing map. Neural Networks 37, 52–65 (2013)
5. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)
6. Vesanto, J.: SOM-based data visualization methods. Intelligent Data Analysis 3(2), 111–126 (1999)
7. Ultsch, A., Siemon, H.P.: Kohonen's self organizing feature maps for exploratory data analysis. In: Proc. of the International Conference on Neural Networks, Paris, pp. 305–308 (1990)
8. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proc. of the 4th Workshop on Self-Organizing Maps, Kyushu, pp. 225–230 (2003)
9. Jain, A.K.: Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
10. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Bookman (2001)
11. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, vol. 1, pp. 28–48. Addison-Wesley (1992)
12. Duda, R., Hart, P., Stork, D.G.: Pattern classification and scene analysis. John Wiley Professional, Wiley (2000)
13. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Self-Organizing Map in Matlab: the SOM Toolbox. In: Proc. of the Matlab DSP Conference, Espoo, pp. 35–40 (1999)
14. Ultsch, A.: Clustering with SOM: U*C. In: Proc. of the Workshop on Self-Organizing Maps, Paris, pp. 75–82 (2005)
15. Karypis, G., Han, E.-H., Kumar, V.: Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 32(8), 68–75 (1999)
16. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010)
17. Costa, J.A.F.: Uma nova abordagem para visualização e detecção de agrupamentos em mapas de Kohonen baseado em gradientes das componentes. Learning and Non Linear Models 9(1), 20–31 (2011)
18. Costa, J.A.F.: Clustering of complex shaped data sets via kohonen maps and mathematical morphology. In: Proc. of the SPIE, Data Mining and Knowledge Discovery, vol. 4384, pp. 16–27 (2001)
19. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)
20. Dougherty, E.R., Lotufo, R.A.: Hands-on Morphological Image Processing. SPIE Publications (2003)
21. Meila, M., Heckerman, D.: An experimental comparison of model based clustering methods. Machine Learning 42(1–2), 9–29 (2001)

Efficient Service Deployment by Image-Aware VM Allocation Strategy

Binbin Huang, Rongheng Lin, Kai Peng, Hua Zou, and Fangchun Yang

State Key Laboratory of Networking Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
{huangbinbin,rhlin,zouhua,fcyang}@bupt.edu.cn,
pkbupt@gmail.com

Abstract. Cloud computing centers face the key challenge of rapidly provisioning diverse virtual machines (VMs) to efficiently deploy services in an elastic and scalable manner. VMs are created by instantiating VM images which encapsulate the required applications or services with their support environment. Thus, the rapid provisioning of VM images is an important factor of efficient service deployment. However, rapid service deployment is determined not only by efficiently VM images fetch but also by request patterns, placement and caching strategies. Thus, in this paper, we explore those effects, and present an image-aware VM allocation strategy to speed up service deployment. Our experiments show that our proposed algorithm largely improves the deployment efficiency. Moreover, the algorithm improves the utilization of cache space and alleviates network traffic.

Keywords: Cloud Computing, Virtual Machine, Virtual Image, Service Deployment, Image-Aware.

1 Introduction

An important feature of cloud computing is elastic resource provisioning, which scales infrastructures behind user's services to meet their actual demands. To allow infrastructure scaling, services which are packed into variable Virtual Machines (VMs) need to be efficiently deployed in IaaS cloud [1]. Thus, the rapid provisioning of diverse VMs is one important factor of efficient service deployment. VMs are created by instantiating VM images that encapsulate the required applications or services with their support environment. And researchers [2][3] reveal that VM creation time is mostly dependent on VM image fetch time. Thus, in fact the availability of VM images plays an important role in the rapid service deployment.

VM image fetch is time consuming [4]. Existing researches [5-7] have proposed different schemes to efficiently fetch VM images, which makes VM images good availability, thereby efficiently deploying services. These researches mainly focus on efficiently deploying services by effectively fetching VM images. However, efficient service deployment is determined not only by efficiently VM images fetch but also by request patterns, placement and caching strategies.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 252–261, 2013.

In this paper, we first abstract a service (such as multi-tier application) into a Virtual network (VN) pattern to guarantee service QoS. Infrastructure providers like Amazon [8] provides SLA where the amount of multi-dimensional resources available are guaranteed to those requested VMs. Network performance guarantees between VMs aren't provided. Enterprise workloads often consist of small requests with specific bandwidth requirements between VMs. Thus, to guarantee specific bandwidth between VMs, we abstract a service request (such as multi-tier application) as a VN pattern, which consists of a set of virtual nodes and virtual edges. Each virtual node requires multi-dimensional server resources for executing the applications. Each virtual edge needs amount of communication bandwidth for the purpose of data exchanging between virtual nodes. Specific bandwidth guarantees between virtual nodes are important to ensure services' QoS. Secondly, we propose an image-aware VN deployment algorithm (IAVND) to mask the time spent in VMs image preparation and speed up VN deployment. Last but not least, we design an efficient cache policy called "Double-TopK" which enhances the cache hit ratio, and reduces VM images transfer time consuming, and alleviates network traffic consuming.

The main contribution of this paper can be summarized as follows: (1) Compare to the conventional single VM deployment, this paper deploys different VNs, which not only makes multi-dimensional server resources available to VMs but also guarantees the communication bandwidth among VMs. Furthermore, it ensures services' QoS very well. (2) We formalize the problem of image-aware VN deployment and present the image-aware VN deployment algorithm to speed up VN deployment. (3) We design an efficient cache policy called "Double-TopK" which enhances the cache hit ratio, and reduces VM images transfer time consuming, and alleviates network traffic consuming.

The rest of this paper is organized as follows. In Section 2, we describe the problem of image-aware VN deployment and formalize it. In section 3, we present the image-aware VN deployment algorithm in detail. In section 4 we describe the simulation setting and present simulation results of our proposed algorithm. Finally, we conclude this paper in section 5.

2 Image-Aware Virtual Network Deployment

In this section, we first present an overview of multi-level VM image cache architecture in which we conduct the image-aware VN deployment. Then we model the problem of image-aware VN deployment. And finally, we define three performance metrics to evaluate the proposed algorithm performance.

2.1 Multi-level VM Image Cache Architecture Overview

Fig.1 presents the scenario where Infrastructure provider maintains N identical clusters, an image repository, a cloud provisioning engine and so on. Each cluster with a cluster manager maintains H physical hosts. The multi-dimensional available resource of a host h is denoted by a vector $C_h = \{C_h^{cpu}, C_h^{memory}, C_h^{disk}\}$, C_h^{cpu},

C_h^{memory} and C_h^{disk} respectively represent the available CPU, available memory, available disk of the host h. All VM images are stored in a central image repository whose size is SP_r. The image repository which is given a set of VM images $I_{tp} = \{I_1, \ldots, I_n\}$ is responsible for image duplicating, image indexing, and version controlling and so on. It also sets up L1 cache and L2 cache to cache the "hot" VM images. L1 with its cache size SP_1 is located in the local disk of each host, while L2 with its cache size $SP_2 (SP_2 = SP_r / N)$ is located in the local disk of each cluster manager.

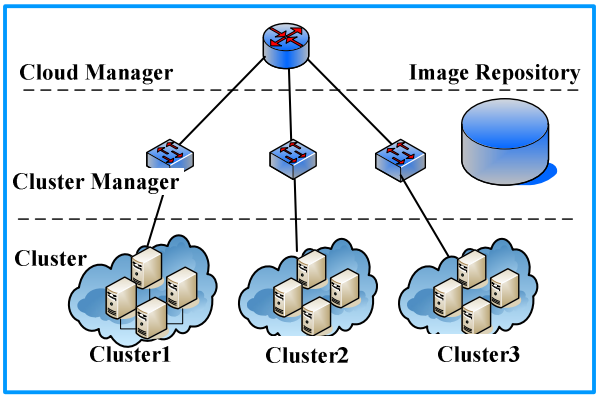


Fig. 1. Multi-level VM image cache architecture overview

2.2 Image-Aware Virtual Network Deployment Model

VN Deployment Time Model. The rapid provisioning of VM images is one important factor of efficient service deployment. Thus, we model the VM images preparation time. Diverse VM images may be needed to service a VN request. The total VMs image preparation time of the i_{th} VN request which is placed in the k_{th} cluster is denoted by T_k^i:

$$T_k^i = \sum\nolimits_{1 \le j \le N_{vm}^i} t_i^j \quad 1 \le k \le N \tag{1}$$

Where t_i^j is the j_{th} VM image preparation time of the i_{th} VN request, and the required multi-dimensional resource of j_{th} VM of the i_{th} VN request is denoted by a vector $c_j = \{c_j^{cpu}, c_j^{memory}, c_j^{disk}\}$, N_{vm}^i is the number of VMs in the i_{th} VN request. When the j_{th} VM image of the i_{th} VN request can be found in L1 cache or L2 cache, it will be fetched directly from L1 cache or L2 cache. In this case, the transfer time t_i^j is short, and can be approximated negligible. Otherwise, it needs to be trans-

ferred from repository to the specific node. The transfer time t_i^j is denoted by $t_i^j = S/B$, where the image size is S MB, and the bandwidth between repository and host is B MB/s. Within a window of R past VN requests, the total VM images preparation time for R VN requests is denoted by $T_{pp}(R)$. Minimizing $T_{pp}(R)$ is one of our objectives.

$$T_{pp}(R) = \sum_{1 \le i \le R} \sum_{1 \le k \le N} x_k^i T_k^i \tag{2}$$

Where $x_k^i = \begin{cases} 1 & \text{if } i_{th} \text{ VN request is placed in } k_{th} \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$, and $x_0^i + x_1^i + ... + x_k^i + ... + x_N^i = 1$.

Cache Policy Model. Given a fixed cache space allocated to L1 and L2 respectively, we adopt the proven best aged caching strategy [4] in L1 and "Double-TopK" in L2. For each cluster, we compute periodically the popularity of VM images in L1, and sort them, and then choose the Top-k of them to cache in L2. At the same time, we remove them from the all L1 caches. This scheme significantly saves storage space and improves the utilization of storage space. We assume the popular VM images in global network are popular in every cluster, therefore we choose the "hot" VM images in each cluster, and sort them, and then select Top-k of them to cache in each L2. Thereby, we call this cache policy "Double-TopK". The L2 composition is also computed at periodic intervals. When the cache is replenished, it is replaced by evicting "cold" VM images to free up resource for new VM images to be cached. To solve VM image replacement problem in L2, we adopt an efficient cache replacement policy called MFRU. We explain in detail MFRU as follow.

MFU computes the popularity of a VM image according to its frequency of use. Within a window of R past requests, the popularity of type I_n VM image can be calculated as follows:

$$U_{I_n} = \sum_{1 \le i \le R,\, I_n \in I_{tp}} \sum_{1 \le j \le N_{vm}^i} r_i^j(I_n) \tag{3}$$

Where r_i^j is a binary variable. It is 1 if the j_{th} VM of the i_{th} VN request requires the type I_n VM image. It is 0 elsewhere. Order of popularity of an VM image is proportional to U_{I_n}. When cache space in L2 is replenished, we can replace "cold" VM images with "hot" VM images according to the order of w_{I_n}:

$$w_{I_n} = \frac{U_{I_n}}{\sum_{I_n \in I_{tp}} U_{I_n}} \tag{4}$$

MFU implicitly assumes a stationary distribution within history window. But as time passed, the popularity of a kind of VM image may fade. The longer the time elapsed since the last request for a kind of VM image is, the lower the likelihood of a

request for that type VM image is. So, MRU is introduced to capture the above temporal variation of request distribution by MFU. MRU computes the popularity of a VM image according to the temporal variation of request distribution. Factoring in temporal aspect, a naïve approach for attenuating the weight w_{I_n} of the type I_n VM image is to reduce the values proportional to the time elapsed since the arrival of a request for this type. The attenuated weight w'_{I_n} is expressed as:

$$w'_{I_n} = \frac{\sum_{1\le i\le R}\sum_{1\le j\le N^i_{vm}} f(j, r^j_i(I_n))}{\sum_{I_n\in I_{tp}}\sum_{1\le i\le R}\sum_{1\le j\le N^i_{vm}} f(j, r^j_i(I_n))} \tag{5}$$

Where $f(x,y) = y * \exp(-x)$. The w'_{I_n} is proportional to the popularity of VM images. L2 is updated according to the order of w'_{I_n}.

3 Performance Metrics

To evaluate the efficiency of our proposed IAVND algorithm, we mainly define four performance metrics as follow.

Define 1: The average provision time $\overline{T_{pp}(R)}$ is defined as the mean of the provision time of R VN requests. It takes T^i_k to dispense the required VM images to different hosts upon receiving a VN request.

$$\overline{T_{pp}(R)} = \left(\sum_{1\le i\le R}\sum_{1\le k\le N} x^i_k T^i_k\right)\Big/ R \tag{6}$$

Define 2: The average network traffic $\overline{SZ_{pp}(R)}$ is defined as the mean of the network traffic which is caused by dispensing the required VM images to different hosts for N VN requests.

$$\overline{SZ_{pp}(R)} = \left(\sum_{1\le i\le R}\sum_{1\le k\le N} x^i_k SZ^i_k\right)\Big/ R \tag{7}$$

The total network traffic of the i_{th} VN request for k_{th} cluster is denoted by $SZ^i_k = \sum_{1\le j\le N^i_{vm}} sz^j_i \quad (1\le k\le N)$. Where sz^j_i refers to the j_{th} VM image size of the i_{th} VN request, N^i_{vm} is the number of VMs in the i_{th} VN request.

Define 3: In order to conduct cost-benefit analysis between the provision time and the unit cache space, we define the time-space overhead ratio r.

$$r = \sum_{1\le k\le N} x^i_k T^i_k \Big/ N * (H * SP_1 + SP_2) \tag{8}$$

Where SP_1 and SP_2 respectively refer to the size for L1 and L2.

Define 4: In order to display the trend of the total occupied cache space with the increase in the total number of requests in different provisioning schemes, we define the total overhead cache space U.

$$U = \sum_{1 \le k \le N} (\sum_{1 \le h \le H} y_k^h SP_1 + SP_2^k) \tag{9}$$

Where $x_k^i = \begin{cases} 1 & \text{if L1 cache of the } h_{th} \text{ host in the is } k_{th} \text{ cluster is used} \\ 0 & \text{otherwise} \end{cases}$. SP_2^k refers to the occupied L2 cache size in the k_{th} cluster.

4 Image-Aware Virtual Network Deploying Algorithm

In this section, we introduce the IAVND algorithm in detail. IAVND mainly includes three stages. At the first stage, we calculate the total VMs image preparation time T_k^i of the i_{th} VN request in k_{th} cluster, and then sort the clusters in ascending order T_k^i. The k_{th} cluster with the minimum T_k^i is selected as the candidate cluster in which the VN request is deployed. At the second stage, it is to find the candidate hosts which meet with the constraints for the virtual nodes in a VN request. At the last stage, it is to find physical paths which satisfy the required bandwidth between pairs of VMs in a virtual network request.

Image-Aware Virtual Network Deployment Algorithm (IAVND)

1: Parameter Initialization: initializes the parameter $M^* = \varnothing$ and $M\sim = M$. M^* and $M\sim$ are respectively the sets of deployed virtual nodes and non-deployed virtual nodes

2: For i_{th} VN request.

3: To compute the time to transfer the needed VM images to different hosts in each cluster according to formula (1).

4: Choosing the k_{th} cluster with the minimum time $\min(T_k^i)$ to deploy the i_{th} virtual network request.

5: While $M\sim \neq \varnothing$ do

6: For the j_{th} VM of the i_{th} virtual network request.

7: If the VM image required by j_{th} VM is first searched in L1 cache, the candidate host must satisfy the conditions as follows:

a) The required VM image by j_{th} VM is in L1 cache.

b) The required multi-dimensional resource c_j by j_{th} VM is no more than the available C_h of the host h.

8: Else if the VM images required by j_{th} VM can be searched in L2 cache, the cluster manager needs to support NFS. And the candidate host must meet with the conditions as follows:

a) The required multi-dimensional resource c_j by j_{th} VM is no more than the available C_h of the host h.

9: Else the VM images required by j_{th} VM cannot be found in L1 or L2 cache, it needs to be transferred from the central repository. And the candidate host must satisfy the same conditions as Step 3.
10: If it succeeds, put the deployed VM into M^*, and remove it from $M\sim$. Otherwise, $k = k + 1$, and go back to Step 4.
11: If $M \sim = \varnothing$, go back to 2.
12: For each virtual edge of the i_{th} virtual network request
13: Find a physical path which satisfies the required bandwidth between pairs of VMs.
14: If not, go back to Step 4.

5 Simulation Results

In order to evaluate the efficiency of our proposed algorithm, we deploy IAVND on an open source simulation framework named CloudSim [9] and compare IAVND scheme with two approaches. The first is called *baseline* where each host only fetches VM image from the central image server. The second is called *local*, where each host uses host-level cache.

We first describe the simulation environment, and then present our simulation results. The experiments focus primarily on the performance comparison of IAVND with two other approaches.

5.1 Simulation Setup

The basic setting is as follows. There is a repository which consists of 21 types VM images (2 types of 3G VM image, 3 types of 9G VM image and 15 types of 6G VM image), and image sizes vary from 3GB to 9GB. There are three clusters in our IaaS. Each cluster consists of 30 hosts. Each host has capabilities of 32GB memory, 32-core Xeon CPU. And no VM images are available at any L1 cache and L2 cache at the beginning of each simulation. As stated in [10], the average network connectivity is fixed at 0.02 in each cluster. The CPU and bandwidth resources of hosts and links are real numbers uniformly distributed between 50 and 100. We model the arrival of VN requests by a Poisson Process with rate 0.05. Each VN request has an exponentially distributed lifetime with an average of 400 time units. The average VN connectivity is fixed at 0.5. The CPU and bandwidth demand of virtual nodes and virtual links are real numbers uniformly distributed between 0 and 50.

5.2 Performance Evaluation

Firstly, we compute the average provision time of 100 VN requests according to Formula 6, and then compare the average provision time using *baseline*, *local* and IAVND. As shown in Fig. 2, we see that the average provision time of IAVND is less than those of two other schemes. The "hot" VM images cached in L1 and L2 are the major contributing factors to performance gain. The benefit from local availability is clearly seen by comparing *local* and *baseline*, while the benefits created by IAVND can be observed by the comparison of IAVND and *local*.

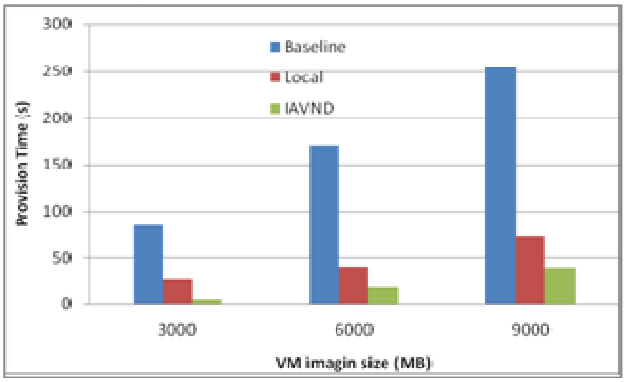


Fig. 2. The average provision time for a VN request using baseline, local, IAVND

We compute the average network traffic of every 20 VN requests according to Formula 7, and compare the average network traffic using these three schemes. As shown in Fig. 3, we see that the average network traffic of IAVND is less than those of other two schemes. That is because the VM images required by VN requests are mostly found in L1 or L2, and thus, there is no need to transfer existing VM images in L1 or L2 from the repository or other clusters.

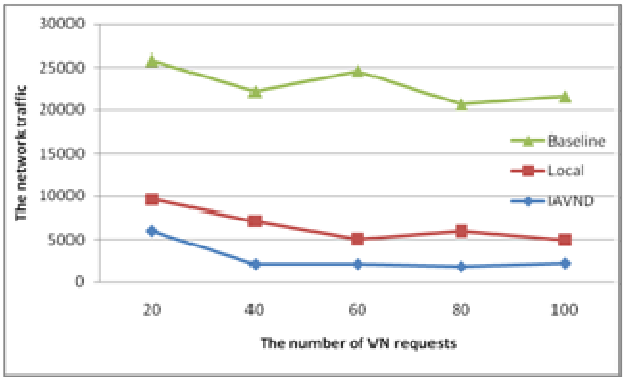


Fig. 3. The average network traffic using baseline, local, IAVND

In order to conduct cost-benefit analysis between the provision time and the cache space, we compute the time-space overhead ratio r according to formula 8, and

compare it using *local* and IAVND. As shown in Fig. 4, we find that the time-space overhead ratio r of IAVND is less than that of *local* with different VM image size, which means that IAVND benefit from less provision at costs of unit cache space.

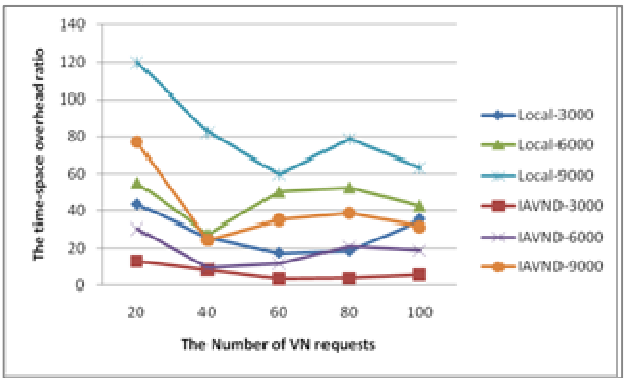


Fig. 4. The time-space overhead ratio using baseline, local, IAVND

In order to display the trend of the total needed cache space with the increase in the total number of requests, we compute the overhead of the total cache space U according to formula 9. As shown in Fig. 5, we find that with the increase of the total number of requests, the growth of the total cache space consumption with IAVND is lower than *local*.

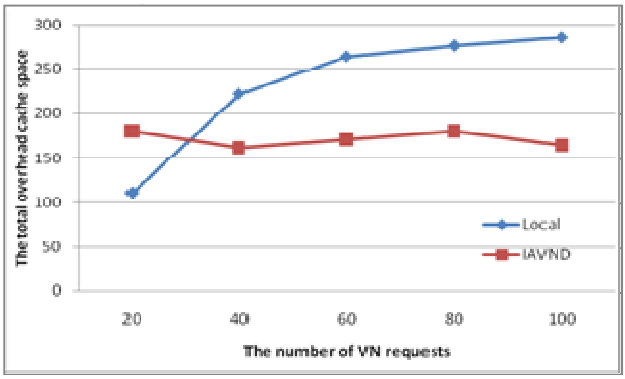


Fig. 5. The total overhead of cache space using baseline, IAVND

6 Conclusion

In this paper, we investigate and solve the rapid service deployment problem. The rapid service deployment is determined not only by efficiently VM images fetch but also by request patterns, placement and caching strategies. To explore those effects, we abstract a service request submitted to infrastructure providers as a VN pattern, and propose IAVND algorithm to deploy efficiently a VN request which guarantees

bandwidth between pairs of VMs. In order to enhance the cache hit ratio, we design an efficient cache policy called "Double-TopK", which always keeps the "hot" VM images in L2, and significantly reduces the time consuming as well as network traffic consuming caused by VM images transfer from the repository to hosts and high-level network traffic among clusters in hierarchical architectures of data center. Experiment results show that IAVND can largely increase the provision efficiency comparing to *baseline* and *local* approaches.

Acknowledgment. This work is supported by the National Natural Science Fund China under Grant No. 2009CB320406, the National 863 High-tech Project of China under Grant No. 2011AA01A102, the Program for New Century Excellent Talents in University (100263), Funds for Creative Research Groups of China (60821001) and State Key Lab of Networking and Switching Technology.

References

1. Fan, P., Chen, Z., Wang, J., Zheng, Z., Lyu, M.R.: Topology-Aware Deployment of Scientific Applications in Cloud Computing. In: IEEE International Conference on Cloud Computing (2012)
2. Zhanga, T., Dua, Z., Chenb, Y., Jic, X., Wang, X.: Typical virtual appliances: An optimized mechanism for virtual appliances provisioning and management. The Journal of System and Software 84, 377–387 (2011)
3. Nishimura, H., Maruyama, N., Matsuoka, S.: Virtual clusters on the fly – fast, scalable, and flexible installation. In: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid, CCGRID 2007, pp. 549–556. IEEE Computer Society, Washington, DC (2007)
4. Emeneker, W., Stanzione, D.: Efficient Virtual Machine Caching in Dynamic Virtual Clusters. In: SRMPDS Workshop, ICAPDS 2007 Conference (Decmber 2007)
5. Jeswani, D., Gupta, M., De, P., Malani, A.: Minimizing Latency in Serving Reqests through Differential Template Caching in a Cloud. In: IEEE International Conference on Cloud Computing (2012)
6. Peng, C., Kim, M., Zhang, Z., Lei, H.: VDN: Virtual Machine Image Distribution Network for Cloud Data Centers. In: IEEE INFOCOM (2012)
7. De, P., Gupta, M., Soni, M., Thatte, A.: Caching Techniques for Rapid Provisioning of Virtual Servers in Cloud Environment. In: IEEE Network Operations and Management Symposium: Short Papers (2012)
8. Amazon virtual private cloud, `http://aws.amazon.com/vpc/`
9. Calheiros, R.N., Ranjan, R., De Rose, C.A.F., Buyya, R.: CloudSim: ANovel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. Technical Report, Grid Computing and Distribution Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne (2009)
10. Ming, J., Baojin, W., Chunming, W., Xiangqing, K., Xiao, M., Min, Z.: Research on Network Virtualization and Virtual Network Mapping Algorithm. Acta Electronica Sinica 39(6), 1315–1320 (2011)

Forecasting Financial Time Series Using a Hybrid Self-Organising Neural Model

Yicun Ouyang and Hujun Yin

School of Electrical and Electronic Engineering, The University of Manchester, Manchester, M60 1QD, UK
yicun.ouyang@postgrad.manchester.ac.uk, hujun.yin@manchester.ac.uk

Abstract. Recently, various variants of the self-organsing map (SOM) have been proposed for modelling and predicting time series. However, most of them are based on lattice structure. In this paper, a hybrid neural model combining neural gas (NG) and mixture autoregressive models is developed for forecasting foreign exchange (FX) rates. It takes advantage of some NG features (i.e. neighbourhood rankings) and incorporates mixture autoregressive models, for effectively modelling and forecasting non-stationary and nonlinear time series. Experiments on FX rates are presented and the results show that the proposed model performs significantly better than other methods, in terms of normalised root-mean-squared-error and correct trend prediction percentage.

Keywords: non-stationary time series, forecasting, FX rates, neural gas, autoregressive models.

1 Introduction

Foreign exchange (FX) rates prediction has been attracting interests from economists, statisticians as well as engineers. Although the task is difficult and challenging, almost impossible in some cases, many researchers and experts have devoted tremendous effort in it due to lucrative returns.

Econometric models such as autoregressive (AR), moving-average (MA) and autoregressive moving-average (ARMA) [1], build a linear relationship among the past points. However, little evidence shows that only linear dependencies exist in financial time series [2]. The nonlinear models, autoregressive conditional heteroscedastic (ARCH) and generalised autoregressive conditional heteroscedastic (GARCH) models, have been often explored for considering volatilities in modelling and forecasting FX rates. However, both ARCH and GARCH models respond the same to either positive or negative shocks, violating a well-known fact that the price of a financial asset responds differently to positive and negative shocks [3].

Artificial neural networks (ANNs) have also been employed for modelling time series. Since the first attempt [4], they have been demonstrated to be competitive alternatives to traditional time series models [5]. Commonly used ANNs include the multilayer perceptrons (MLP), support vector machines (SVM) and the self-organising map (SOM). In addition, recurrent networks such as the echo state networks (ESN) have been reported performing well in such forecasting tasks [6].

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 262–269, 2013.

The standard SOM, also termed the vector SOM (VSOM) in time series modelling [7], groups consecutive points into segments of fixed length as the input vectors for training the SOM. The temporal Kohonen map [8] and its improved version, recurrent SOM (RSOM) [9], introduce previous activations in computing current activations. In the recursive SOM (RecSOM) [10], temporal information was captured with the activation being used as another input homogeneous to the original input.

A mixture autoregressive model, termed self-organising mixture autoregressive (SOMAR) [13], learns local regressive models via a self-organising map. It has been demonstrated that the SOMAR model obtained improved performance in modelling and predicting on a variety of time series over regressive and other SOM-based models.

The rest of this paper is organised as follows. Section 2 describes the proposed model in details. The experiments and results are presented and discussed in section 3. The final section concludes the study.

2 Methodology

The neural gas (NG) [11] is a variant of SOM, in which the topological structure of the neurons is not fixed as lattice. Instead, it has an adaptive structure and updates the weights of reference vectors according to neighbourhood rankings. The closest reference vectors to the input vector, the second closest, third closest and so on, are adapted according to the distances to the input vector. The argument is that data distributes rarely in a lattice space but often in a free space. Therefore, NG can capture the distribution better. The weight adaptaion rule is

$$\mathbf{w}_{i_k}(t+1) = \mathbf{w}_{i_k}(t) + \epsilon e^{-k/\lambda}(\mathbf{x} - \mathbf{w}_{i_k}(t)) \tag{1}$$

where i_0 and i_{N-1} denote the index of the closest and the most distant reference vectors to the input vector $\mathbf{x}$, $k = 0, ..., N-1$ is the ranking of reference vectors, both the learning rate or adaption step size ϵ and neighbourhood range λ decay with time.

NG can achieve more robust convergence with faster speed and smaller distortion errors than the standard SOM [11]. The proposed model employs the structure of NG to organise local regressive models.

2.1 Self-Organising Mixture Autoregressive Model

Ni and Yin proposed a method SOMAR and applied it to FX rates prediction [13]. The SOMAR model employs the sum of autocorrelation coefficients (SAC) as the similarity measure, to calculate the distance between input and reference vectors. It is defined as

$$v(i) = \sum_{j=-m}^{m} \|R_i(j)\| \tag{2}$$

where $R_i(j)$ is an estimate of the autocorrelation coefficient at lag j, i is the index of local model and m is the length of modelling error.

The model was reported working well in converging to underlying local regressive models, outperforming conventional statistical models and SOM-based methods in modelling and predicting time series. However, the structure of SOMAR is confined on a lattice as in the standard SOM.

2.2 Neural Gas Mixture Autoregressive

As the mixture autoregressive (MAR) framework is effective for non-stationary and nonlinear time series, it is used as the basis in the proposed method. Instead of using the SOM to realise the mixture framework, the NG is employed for more flexible structures. The proposed method is termed as neural gas mixture autoregressive (NGMAR) model. The input vectors are used in small batches, similar to the SOMAR; and the SAC (sum of autoregressive coefficients) is also used as the similarity measure. Assuming the size of batch is m, the input batch $\mathbf{x}(t)$ and the modelling error $\mathbf{e}_i(t)$ at node i are defined as

$$\mathbf{x}(t) = \{x_j(t)\},\ j = 1, 2, ..., m \tag{3}$$

$$\mathbf{e}_i(t) = \mathbf{x}(t) - \mathbf{w}_i^T \mathbf{x}^{(p)}(t-1) \tag{4}$$

where p and $\mathbf{w}_i$ are the order and parameters of local AR model i, respectively. Then the autocorrelation coefficient $R_i(k)$ at lag $k(k < m)$ is obtained by

$$R_i(k) = \frac{1}{m\sigma^2} \sum_{s=0}^{m-k-s} (\mathbf{e}_i(t-s) - u)(\mathbf{e}_i(t+k-s) - u) \tag{5}$$

where m, μ and σ^2 are the length, mean and variance of the modelling error. The best matching unit (BMU) is selected with the minimum SAC value. Then,

$$SAC = \sum_{k=-p}^{p} \|R_i(k)\| \tag{6}$$

In the NGMAR model, the reference vectors are adapted according to their distances to the input and the SAC rankings as,

$$\mathbf{w}_{i_k}(t+1) = \mathbf{w}_{i_k}(t) + \epsilon \mathbf{e}_{i_k}^{-k/\lambda}(\mathbf{x}(t) - \mathbf{w}_{i_k}(t)) \tag{7}$$

where $\mathbf{x}(t) = \{x_1(t), x_2(t), ..., x_m(t)\}$ is the input batch, $\mathbf{w}$ is the weight of reference vector, k is the neighbourhood ranking of the SAC.

3 Experimental Results

We used three main FX rates to evaluate the prediction performance of the NGMAR model. The mean values and standard variances of the NGMAR and various models were obtained for comparison. p values of the t-test were also calculated for validating the significance.

Algorithm 1. The training procedure of NGMAR

Require: small batches of input vectors from sample time series;
Ensure: the reference vectors representing for weights of neurons;
1: Build the input vectors $\mathbf{x}(t)$ from the training time series;
2: Initialise the weight matrix $\mathbf{w}_i$, and specify training parameters learning rates l and training iterations N;
3: **for** $t = 1$ to N **do**
4: **for** each i of all neurons **do**
5: Compute individual error $err(t, i)$ of neuron i and input $\mathbf{x}(t)$ on Eq. (4);
6: Compute the SAC value $SAC(i)$ on Eq. (6);
7: **end for**
8: Compute rank of neurons and build RANK
9: **for** each i of all neurons **do**
10: Find the position $r(i)$ of neuron i in RANK
11: Update the weights $\mathbf{w}_i$ of neuron i on Eq. (7);
12: **end for**
13: **end for**
14: **return** the weight matrix $\mathbf{w}_i$;

3.1 Foreign Exchange Rates

The daily closing prices of three exchange rates (British Pound vs. US Dollar, Euro and JP Yen, respectively) from 2001 to 2012 (retrieved from the PACIFIC exchange rate service) were used in the experiments. Fig. 1 plots these three FX rates. The 10-fold cross-validation was applied. That is, each dataset (3000 points) was divided into ten segments, and nine of them were randomly chosen as the training set and the remaining one as the test set. The sliding window size of input vectors was selected as 12 according to the Akaike information criterion (AIC) and Bayesian information criterion (BIC) results. All concerned models and methods were performed for one-step-ahead prediction.

As for each model/method used in the experiments, its predictive ability was described through an out-of-sample prediction scheme. As reviewed in [14] [15], there are a number of measures of predictive accuracy. The following two were employed in the experiments.

NRMSE of Predicted Rate: The normalised root mean squared error (NRMSE) [16] [17] between the actual and predicted exchange rates in the test set.

Predicted Return (%): The correct prediction percentage (CPP) [7] [13] of the FX rate return (sign of $ln\frac{x_{t+1}}{x_t}$) is used to describe how many percentages of the prediction is made in the correct direction, i.e. the percentage of the predicted returns having the same sign as the actual returns.

The actual and predicted test FX rates by NGMAR are plotted in Figs. 2-4 and their quantitative evaluations are shown in Table 1. T-test was used to test the significance of the results. It can be seen that the NGMAR model outperformed SOM-based models and other neural networks. The significant tests (p-value) show that the improvements by the NGMAR over all other methods

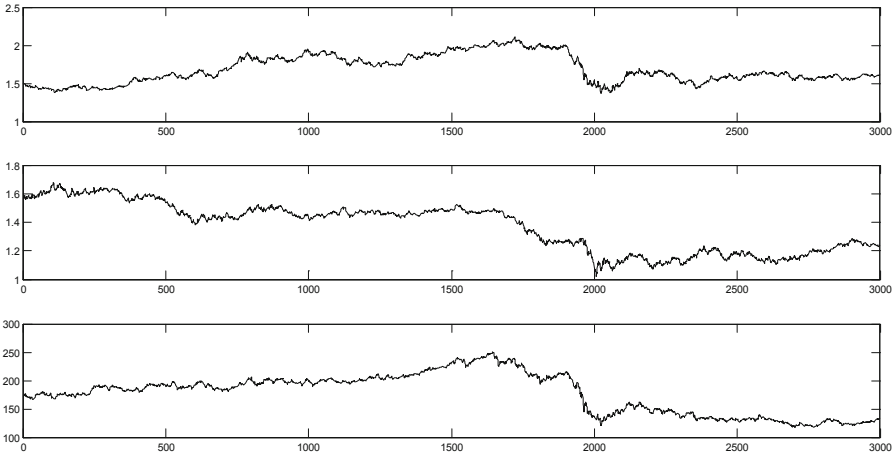


Fig. 1. GBP vs USD, EUR and JPY daily closing prices

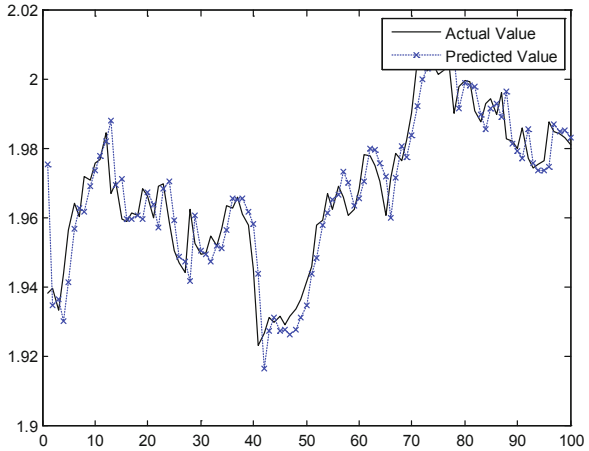


Fig. 2. Actual and predicted test series for GBP/USD time series

are significant, about 1.4-6.5%. The closest one is the SOMAR, the predecessor of NGMAR. The two have about 1-2% differences in CPP. In the experiments, NGMAR and SOMAR used the same similarity measure SAC; however, NGMAR obtained better results than SOMAR, and this can be explained by the dynamic structure and neighbourhood ranking of NG. The structure of NG is varied according to neighbourhood rankings, which make weights updated more accurately. It is also observed that NGMAR significantly outperformed NG, and this can be explained by the usefulness of the new similarity measure SAC.

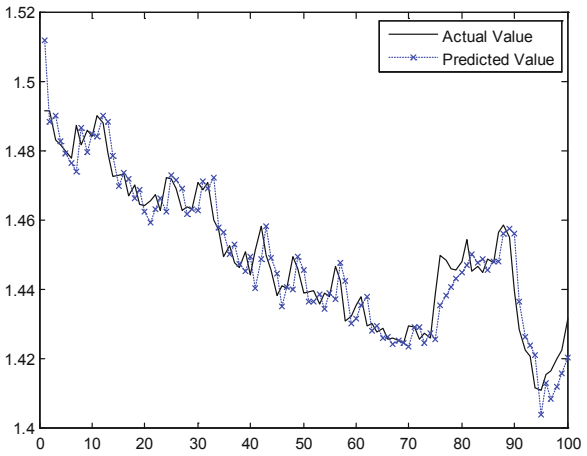


Fig. 3. Actual and predicted test series for GBP/EUR time series

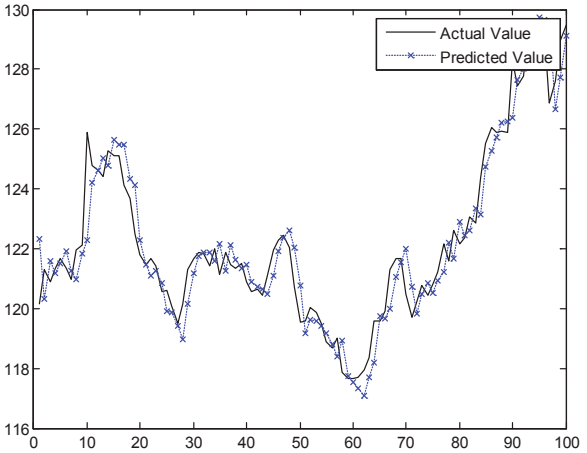


Fig. 4. Actual and predicted test series for GBP/JPY time series

Table 1. Prediction errors (NRMSE), correct prediction precentage (%) and significance (p) of various models on three FX rates

GBPvs	US Err	US CPP	US p	EU Err	EU CPP	EU p	JP Err	JP CPP	JP p
VSOM	0.1121	50.32	**4.7e-7**	0.0838	51.55	**0.0071**	0.1118	50.98	**7.3e-4**
VQTAM	0.1152	50.69	**4.5e-6**	0.0795	51.14	**0.0015**	0.1056	50.72	**0.0012**
RSOM	0.1014	51.20	**3.5e-5**	0.0803	51.61	**0.0080**	0.1082	51.98	**0.0015**
RecSOM	0.1047	51.07	**7.2e-5**	0.0771	51.65	0.0173	0.1035	51.57	**5.7e-4**
SOAR	0.1006	51.08	**6.8e-5**	0.0685	51.48	**7.1e-4**	0.0878	51.33	**4.3e-4**
SOMAR	0.0812	55.42	**0.0098**	0.0517	53.26	0.0151	0.0755	53.92	0.0214
NG	0.0959	51.89	**1.2e-5**	0.0736	51.86	**0.0041**	0.1021	52.11	**0.0073**
NGMAR	**0.0671**	**56.81**	——-	**0.0379**	**55.36**	——-	**0.0605**	**55.70**	——-
MLP	0.1100	50.52	**3.1e-7**	0.0787	50.89	**1.3e-4**	0.1045	52.92	**0.0097**
SVM	0.1058	50.79	**4.4e-6**	0.0792	50.81	**1.1e-4**	0.0988	53.01	**0.0087**
ESN	0.1001	51.72	**5.0e-4**	0.0761	51.10	**4.2e-4**	0.0789	52.78	**0.0085**

4 Conclusions

In this paper, a new approach, NGMAR, has been proposed to model and forecast time series. It organises the local mixture autoregressive models as neural gas and uses them to capture and describe the behaviour of time series. The experimental results show that this new neural model is more accurate than SOM-based models and other neural networks for predicting non-stationary FX rates time series. They also show that the improved performance of NGMAR model is significant under the given criteria.

References

1. Tsay, R.S.: Conditional heteroscedastic time series models. Journal of the American Statistical Association 82(398), 590–604 (1987)
2. Huang, W., Lai, K.K., Nakamori, Y., Wang, S.: Forecasting foreign exchange rates with artificial neural networks: A review. International Journal of Information Technology & Decision Making 3(1), 145–165 (2004)
3. Tsay, R.S.: Analysis of Financial Time Series. Wiley-Interscience, Hoboken (2005)
4. Lapedes, A., Farber, R.: Nonlinear signal processing using neural networks: Prediction and system modelling. Technical Report, LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (1987)
5. Zhang, G., Eddy Patuwo, B., Hu, Y.M.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14(1), 35–62 (1998)
6. Crone, S.F., Hibon, M., Nikolopoulos, K.: Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. International Journal of Forecasting 27(3), 635–660 (2011)
7. Yin, H., Ni, H.: Generalized self-organizing mixture autoregressive model for modeling financial time series. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 577–586. Springer, Heidelberg (2009)
8. Chappell, G.J., Taylor, J.G.: The temporal Kohonen map. Neural Networks 6(3), 441–445 (1993)

9. Varsta, M., Milln, J.D.R., Heikkonen, J.: A recurrent self-organizing map for temporal sequence processing. In: Proceedings of the 7th International Conference on Artificial Neural Networks, Lausanne, Switzerland, pp. 421–426 (1997)
10. Voegtlin, T.: Recursive self-organizing maps. Neural Networks 15(8-9), 979–991 (2002)
11. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: "Neural-gas" network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks 4(4), 558–569 (1993)
12. Lampinen, J., Oja, E., SCIA: Self-organizing maps for spatial and temporal AR models. In: Proceedings of the 6th Scandinavian Conference on Image Analysis (SCIA), Helsinki, Finland, pp. 120–127 (1989)
13. Ni, H., Yin, H.: A self-organising mixture autoregressive network for FX time series modelling and prediction. Neurocomputing 72(16-18), 3529–3537 (2009)
14. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443–473 (2006)
15. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting 22(4), 679–688 (2006)
16. 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)
17. Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions on Neural Networks and Learning Systems 23(5), 787–799 (2012)

A Novel Diversity Maintenance Scheme for Evolutionary Multi-objective Optimization

Sen Bong Gee, Xin Qiu, and Kay Chen Tan

Department of Electrical and Computer Engineering
National University of Singapore, Singapore
{a0039834,qiuxin,eletankc}@nus.edu.sg
http://www.nus.edu.sg

Abstract. Recently, decomposition-based multi-objective evolutionary algorithm (MOEA/D) has received increasing attentions due to its simplicity and decent optimization performance. In the presence of the deceptive optimum, the weight vector approach used in MOEA/D may not be able to prevent the population traps into local optimum. In this paper, we propose a new algorithm, namely Diversity Preservation Multi-objective Evolutionary Algorithm based on Decomposition (DivPre-MOEA/D), which uses novel diversity maintenance scheme to enhance the performance of MOEA/D. The proposed algorithm relaxes the dependency of the weight vector approach on approximated ideal vector to maintain diversity of the population. The proposed algorithm is evaluated on CEC-09 test suite and compared the optimization performance with MOEA/D. The experiment results show that DivPre-MOEA/D can provide better solutions spread along the Pareto front.

Keywords: Diversity-preservation, multi-objective evolutionary algorithm, decomposition.

1 Introduction

Over the past few decades, evolutionary algorithms (EAs) have been widely used to solve the multi-objective optimization problems. A multi-objective optimization problem (MOP) involves two or more conflicting objectives that need to be optimized simultaneously. Since MOPs have more than one objective function, the optimal solutions are defined as the solutions which provide the best trade-off between different objective functions. As it is impractical to find all trade-off solutions, the task of evolutionary multi-objective (EMO) algorithm is to approximate the Pareto optimal front (PF) which is formed by the best trade-offs solutions in objective space.

Multi-objective evolutionary algorithm (MOEA) is an algorithm that generates a set of solutions approximating the PF of an MOP. To achieve a high quality of approximation, the generated solutions must be close to and well spread along the PF. The closeness of the approximated Pareto objective vectors to the PF is

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 270–277, 2013.

related to the convergence properties of an algorithm whereas the distribution of the approximated Pareto objective vectors is related to the diversity properties of an algorithm. The diversity and convergence properties are often conflicting to each other as strong convergence properties most likely result in poor diversity and vice versa. Therefore, a good balance between these two properties is important in the design of an MOEA.

Diversity properties of an MOEA generally can be improved by either diversity enhancement or diversity preservation mechanism. The former improves the diversity of solutions by producing more diverse offspring population whereas the latter maintains the diversity of solutions by selecting more diverse parents. The diversity enhancement mechanism is typically applied within the reproduction stage [1, 7, 13, 14]. For the diversity preservation mechanism, certain heuristics are embedded in the selection operator to maintain the diversity of the whole population. According to [2], diversity preservation techniques of MOEAs can be categorized into Weight Vector Approach [6, 15], Fitness Sharing/Niching Approach [4, 5], Crowding/Clustering [3, 11], Restricted Mating [10], and Relaxed Dominance [8]. In this paper, we propose a new diversity preservation method which is different from the other existing methods. The proposed method quantifies the diversity loss using a metric and restricts the maximum allowable diversity loss to preserve the diversity of population over the evolution process.

The rest of the paper is organized as follows. Section 2 provides background and MOP basics. Section 3 introduces the proposed method which uses the concept of maximum relative diversity loss (MRDL). Section 4 presents the experiment results and compares the proposed algorithm to MOEA/D with SBX crossover (MOEA/D-SBX) and Differetial Evolution (MOEA/D-DE) operator. Conclusions are drawn in Section 5.

2 Background and Definitions

A multi-objective optimization problem (MOP) consists of more than one objective function that need to be optimized simultaneously. Mathematically, an MOP can be expressed as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \ldots \ f_m(\mathbf{x})]^T \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where f_i is the i-th objective function; m is the number of objective functions; $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ is the objective vector; n is the number of decision variables; $\mathbf{x} \in \mathbb{R}^n$ is the decision vector and $\Omega \subset \mathbb{R}^n$ is the feasible decision space. Without loss of generality, a minimization problem is considered here. Generally, Ω can be described by

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n | g_j(\mathbf{x}) \leq 0, h_k(\mathbf{x}) = 0, j = 1, \ldots, J \text{ and } k = 1, \ldots, K\}$$

where g_j is the j-th inequality constraints out of total J inequality constraints and h_k is the k-th equality constraint out of total K equality constraints. Since

(1) involves more than one objective function, they might be no single point in Ω that minimizes all the objectives simultaneously. Therefore, concepts of Pareto dominance and Pareto optimality are needed to define the objective and decision vector set that provides the best trade-offs between different objective functions [12].

3 Diversity Preservation Multi-objective Evolutionary Algorithm based on Decomposition

Diversity Preservation Multi-objective Evolutionary Algorithm (DivPre-MOEA/D), as the name suggests, is a decomposition-based multi-objective evolutionary algorithm (MOEA/D) that enhances the diversity preservation mechanism. This algorithm improves the diversity properties of MOEA/D by restricting the maximum allowable diversity loss. To measure the diversity loss over two consecutive generations, a novel diversity measurement technique, maximum relative diversity loss (MRDL) method, is proposed.

DivPre-MOEA/D prevents the existence of duplicated solutions in the approximated Pareto solutions. This property is desirable as duplicated solutions not only provide no information for PF approximation, but also may waste the fitness evaluations during the search process. In the worst-case scenario, when all the approximated Pareto solutions sink into one local minimum, there is little chance for the population jumps out of the local minimum as crossover operator generates identical (or similar) offspring, the exploration and exploitation of the algorithm depend solely on mutation strategy. Therefore, there is a need to ensure that there are no duplicated approximated Pareto solutions during the search process.

3.1 Maximum Relative Diversity Loss

Relative diversity loss (RDL) is a diversity measurement quantity that indicates the amount of diversity loss of an individual solution between two consecutive generations with respect to a particular pair of solutions. To compute this quantity, at least one pair of parent and offspring solution is needed and the offspring solution must dominate the parent solution.

To illustrate the way of computing RDL, a simple example is given. Let assume that we have a parent-offspring pair, B and C, as shown in Fig. 1. For the ease of illustrations, B is named reference parent whereas C is called reference offspring. The RDL of parent A to offspring D (with respect to B and C) is defined as

$$\Gamma^{\mathbf{b}\rightarrow\mathbf{d}}_{\mathbf{d}_{\mathrm{conv},1}} = \frac{\triangle ABC}{\triangle BCD} \tag{2}$$

where $\triangle ABC$ and $\triangle BCD$ are the area of triangle ABC and BCD respectively. If there are k parent-offspring reference pairs, the maximum relative diversity loss (MRDL, $\Gamma^{\mathbf{p}\rightarrow\mathbf{c}}$) of parent A to offspring D is defined as

$$\Gamma^{\mathbf{p}\rightarrow\mathbf{c}} = \max_{i=1,\ldots,k} \Gamma^{\mathbf{p}\rightarrow\mathbf{c}}_{\mathbf{d}_{\mathrm{conv},i}} \tag{3}$$

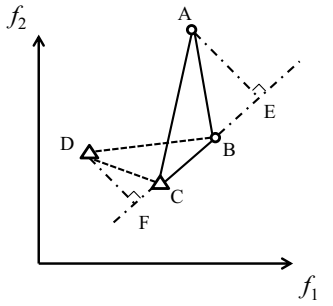


Fig. 1. Parent B and offspring C form a reference pair which will be used to compute the relative diversity loss of parent A to offspring D

The magnitude of MRDL is governed by the highest RDL among the k parent-offspring pairs. Below are some properties and implications of the MRDL:

1. MRDL must be non-negative, i.e. MRDL $\notin \mathbb{R}^-$.
2. If an offspring is identical to any element in the reference offspring set, the MRDL will be infinite.
3. If a parent is identical to any element in the reference parent set, the MRDL will be zero.
4. Geometrically, MRDL represents the ratio of parent and offspring perpendicular distance to the line of reference parent to reference offspring (ratio of AE to DF).
5. High value of MRDL indicates the existence of similar solution (the similarity is based on Euclidean distance of two vectors in objective space) in the reference offspring set.

During the search process, convergence properties of an algorithm exerts selection pressure in a direction normal to the trade-offs region whereas diversity properties of the algorithm control the spread of the solutions. If BC is the approximated direction of the convergence force, the ratio of AE to DF (MRDL) can be a quantity that represents the reduction (increase) of a particular solution's spread. We will use this quantity to indicate a particular solution's diversity loss (gain) during two consecutive generations.

3.2 Estimation of the Convergence Direction

To compute MRDL, we need to know the convergence direction. We define loosely convergence direction as a direction from any point in the feasible objective space to the closest point on PF. To estimate the convergence direction, we must extract the information from the parent and offspring population as they are the only resources that provide information during the search. One simple way to estimate the convergence direction is to use one parent and one offspring

solution, and the offspring solution must dominate the parent solution. In addition, the parent solution must be the closest solution to the offspring solution (in the objective space) among the parent population.

3.3 Main Algorithm

DivPre-MOEA/D incorporates the concept of MRDL into selection operator of MOEA/D. The flow of the algorithm is shown in *Algorithm 1.* Tchebycheff Approach [12] is used to decompose the MOP into N scalar optimization subproblems, where N is the population size of the algorithm. In the selection stage, the proposed algorithm first checks whether the offspring solution is better than any parent in the population. If the offspring dominates any of the parents, the algorithm calculates the MRDL of replacing the parent solution with the offspring solution. If the MRDL is lower than the preset threshold, γ, offspring solution replaces parent solution. Otherwise, the parent solution remains in the population. This characterizes one of the major differences between DivPre-MOEA/D and MOEA/D. In DivPre-MOEA/D, a non-dominated offspring is not necessary to replace the parent solution. Although this characteristic may raise some doubts about the algorithm's convergence at first sight, the empirical results show that the convergence of DivPre-MOEA/D is no worse than MOEA/D. Before the parent-offspring replacement happened, the offspring objective vector is stored in the reference offspring set whereas the closest parent objective vector is stored in the reference parent set. These two vector sets are used to approximate the convergence direction which is one of the most important element in the calculation of MRDL.

4 Experiment Results

This section presents the optimization performance of DivPre-MOEA/D in terms of inverted generational distance (IGD). IGD values is calculated based on 10^3 and 10^4 Pareto optimal points for 2-objective and 3-objective problems, respectively. CEC-09 test suite with recommended settings is used to evaluate the algorithm's performance [16].

Table 1. Parameter Settings for Experiments

Parameters	**Values**
Population size, N	100 or 300 (2 or 3 objectives)
Total number of generation	500
Total number of fitness evaluation	5×10^4 or 1.5×10^5
Neighbourhood size, T	20
Distribution index in SBX, η_c	20
Distribution index in mutation, η_m	20
Mutation rates, p_m	$1/n$ (n: decision variables)
Hybrid recombination strategy parameter, γ	2
Maximum allowable MRDL, γ	20

Algorithm 1. Diversity Preservation Multi-objective Evolutionary Algorithm Based on Decomposition

Input:
MOP (1); A stopping criterion; N: Population size; γ: Maximum allowable MRDL
T: The number of the weight vectors in the neighbourhood of each weight vector
Output:
Approximated Pareto Front ($\mathbf{f}^1, \ldots, \mathbf{f}^N$); Approximated Pareto Solutions ($\mathbf{x}^1, \ldots, \mathbf{x}^N$)
Step 1 ➢ Initialization:

Step 1.1 Generate weight vectors. Find the T closest weight vectors (in terms of Euclidean distance) for each vector. Set $B(i) = i_1, \ldots, i_T$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.
Step 1.2 Randomly generate an initial population, $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N$. Set $\mathbf{f}^i = \mathbf{f}(\mathbf{x}^i)$.
Step 1.3 Initialize approximated ideal vector $\mathbf{z}$ by setting $z_k = \min_{j=1,\ldots,N} f_k^j$ where $k = 1, \ldots, m$.

Step 2 ➢ Update: Set $P = C = \emptyset$. For $i = 1, \ldots, N$, do

Step 2.1 Reproduction: Randomly select two parent from the population and used the crossover operator to produce a new solution. Apply polynomial mutation to the new solution to produce $\mathbf{y}$.
Step 2.2 Update of z: Evaluate $\mathbf{y}$ to get $\mathbf{f}(\mathbf{y})$. If $f_j(\mathbf{y}) < z_i$ for any $j \in \{1, \ldots, m\}$, set $z_i = f_j(\mathbf{y})$
Step 2.3 Selection: Declare a set $D = \{1, \ldots, N\}$ and permute the sequence in the set. For each $j = 1, \ldots, N$, set $k = D_j$, if $g^{te}(\mathbf{y}|\lambda, \mathbf{z}) < g^{te}(\mathbf{x}^k|\lambda, \mathbf{z})$, use *Algorithm 2* to get the MRDL. If MRDL$< \gamma$, set $\mathbf{x}^k = \mathbf{y}$, $\mathbf{f}^k = \mathbf{f}(\mathbf{y})$, $C = C \cup \{\mathbf{f}(\mathbf{y})\}$, $P = P \cup \{\mathbf{w}\}$, where $\mathbf{w}$ is the nearest parent objective vector to $\mathbf{y}$, and then directly go to next step.

Step 3 ➢ Stopping Criterion: If the stopping criterion is satisfied, stop the process, output $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ and $\{\mathbf{f}^1, \ldots, \mathbf{f}^N\}$. Otherwise, go to **Step 2**.

4.1 Comparative Studies

All the above-mentioned algorithms are implemented in C++. The parameter settings of algorithms are shown in Table 1. The effects of introducing the MRDL into the MOEA/D are studied here as shown in Table 2. MOEA/D-DE [9] (MOEA/D-SBX [15]) denotes the MOEA/D with DE mutation operator (SBX operator). The values in the same row of MOEA/D-DE (MOEA/D-SBX) are the average of IGD values (over 30 independent runs) under different test problems. "(+) MRDL" denotes a particular algorithm combined with the proposed MDRL control mechanism. Similarly, the values in the same row of "(+) MRDL" are the average IGD values (over 30 independent runs) under different test problems. The row starts with "Effect" represents the effect of implementing a particular mechanism or strategy. The positive sign in the "Effect" row represents the decrease of the IGD mean and standard deviation (positive effect) whereas the negative sign represents the increase of the IGD mean and deviation (negative effect).

4.2 Discussions

From Table 2, we can observe that MRDL control mechanism generally has a positive effect to MOEA/D under CEC-09 test problems. This is because CEC-09 test problems favor the MOEA with strong diversity properties whereas the MRDL control mechanism can reduce the loss of diversity during the search.

Algorithm 2. Compute the Maximum Relative Diversity Loss

Input:
P: The reference parents objective vector set.
C: The reference offspring objective vector set.
$\mathbf{x}$: The parent objective vector.
$\mathbf{y}$: The offspring objective vector.
Require: $|P| = |C|$
Output:
g: Maximum Relative Diversity Loss

```
if P = ∅ then
    g = 0
    return
else
    s = |P|
    max = 0
    for i = 1 → s do
        Calculate r = △xP_iC_i / △yP_iC_i
        if r > max then
            max = r
        end if
    end for
    g = max
    return
end if
```

Table 2. Compare the Effects of MRDL and HRS (CEC-09 Test Suite)

		UF1	UF2	UF3	UF4	UF5	UF6	UF7	UF8	UF9	UF10
MOEA/D-DE	Average	0.04753	0.04273	0.15146	0.08661	0.7668	0.43858	0.10179	0.09108	0.10654	0.58264
	Std. Dev.	0.03733	0.03171	0.06881	0.01038	0.13277	0.22031	0.1646	0.0124	0.04522	0.07164
(+) MRDL	Average	0.04034	0.02598	0.05789	0.08227	0.79702	0.359	0.03109	0.09345	0.07721	0.98578
	Std. Dev.	0.00823	0.00277	0.02086	0.00963	0.13334	0.11316	0.03421	0.00969	0.02951	0.15556
Effect	Average	+	+	+	+	-	+	+	-	+	-
	Std. Dev.	+	+	+	+	-	+	+	+	+	-
MOEA/D-SBX	Average	0.15695	0.0642	0.30664	0.05605	0.43942	0.43745	0.35331	0.14799	0.13399	0.29365
	Std. Dev.	0.06531	0.03113	0.02993	0.00342	0.08413	0.15079	0.15506	0.03579	0.0624	0.13036
(+) MRDL	Average	0.09371	0.03618	0.19871	0.0537	0.47547	0.30208	0.23432	0.12274	0.1012	0.24653
	Std. Dev.	0.02599	0.00803	0.03851	0.00193	0.1044	0.09997	0.13708	0.02166	0.04096	0.09301
Effect	Average	+	+	+	+	-	+	+	+	+	+
	Std. Dev.	+	+	-	+	-	+	+	+	+	+

Comparing the mean values of different test problems, it can be observed that MRDL control mechanism lowers the IGD means of all test problems except UF5 and UF10. The table also shows that the MRDL control mechanism generally reduces the performance variation as most of the proposed method's standard deviations are smaller than the typical MOEA/D.

5 Conclusions

In this paper, we have proposed the novel online diversity measurement technique, maximum relative diversity loss (MRDL), which is used to preserve the diversity of the population. The proposed diversity measurement method can also be used to control the operator parameter in the hope to improve the algorithm's optimization performance.

References

1. Chow, C., Yuen, S.: A multiobjective evolutionary algorithm that diversifiles population by its density. IEEE Transactions on Evolutionary Computation 16(2), 149–172 (2012)
2. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary algorithms for solving multi-objective problems, vol. 5. Springer (2007)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Fonseca, C., Fleming, P.: Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In: First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA (Conf. Publ. No. 414), pp. 45–52. IET (1995)
5. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application. Erlbaum Associates Inc., pp. 41–49 (1987)
6. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 28(3), 392–403 (1998)
7. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation 14(6), 985–998 (2010)
8. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evolutionary Computation 10(3), 263–282 (2002)
9. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)
10. Lu, H., Yen, G.: Rank-density based multiobjective genetic algorithm. In: IEEE Congress on Evolutionary Computation, vol. 1, pp. 944–949. IEEE (2002)
11. Mahfoud, S.: Crowding and preselection revisited. Parallel Problem Solving from Nature 2, 27–36 (1992)
12. Miettinen, K.: Nonlinear multiobjective optimization, vol. 12. Springer (1999)
13. Qu, B., Suganthan, P.: Multi-objective di erential evolution with diversity enhancement. Journal of Zhejiang University-Science C 11(7), 538–543 (2010)
14. Tan, K., Yang, Y., Goh, C., Lee, T.: Enhanced distribution and exploration for multiobjective evolutionary algorithms. In: IEEE Congress on Evolutionary Computation, vol. 4, pp. 2521–2528. IEEE (2003)
15. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
16. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. Tech. rep., University of Essex, Colchester, UK and Nanyang Technological University, Singapore (2008)

Adaptive Differential Evolution Fuzzy Clustering Algorithm with Spatial Information and Kernel Metric for Remote Sensing Imagery*

Ailong Ma, Yanfei Zhong**, and Liangpei Zhang

State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, P.R. China
zhongyanfei@whu.edu.cn

Abstract. In this paper, an adaptive differential evolution fuzzy clustering algorithm with spatial information and kernel metric for remote sensing imagery, namely KADESFC, is proposed. In KADESFC, the clustering problem is transformed into an optimization problem, which minimizes a proposed kernelized objective function with an adaptive spatial constraint term. Differential evolution algorithm is utilized to optimize the kernelized objective function, which uses several differential evolution operators. Experimental results on two remote sensing images show that the proposed algorithm is promising compared with several traditional clustering algorithms.

Keywords: Fuzzy clustering, differential evolution , kernel metric, adaptive, remote sensing.

1 Introduction

Clustering is one of the most important techniques in remote sensing image processing. The aim of the remote sensing clustering is to partition a given image into groups such that pixels in the same group are as similar to each other as possible while those ass igned to different groups are dissimilar [1]. However, due to the characteristics of the remote sensing image, there are still some problems in remote sensing image clustering. Firstly, some isolated pixels may emerge because of noise, outliers or mixed pixels. This may result from the disregard of the spatial information. Ahmed *et al.* propose FCM_S [2], aiming at incorporating spatial information by modifying the objective function of fuzzy c-means (FCM). However, FCM_S is time-consuming because of the computation of spatial neighborhood term in each iteration. Chen and Zhang [3] reduce the computational complexity of FCM_S by introducing the mean-filtered image named FCM_S1. One common drawback of the above methods is that they both need to determine a parameter α to control the trade-off

* This work was supported by National Natural Science Foundation of China under Grant No. 41371344, and A Foundation for the Author of National Excellent Doctoral Dissertation of P.R. China (FANEDD) under Grant No. 201052.

** Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 278–285, 2013.

between robust to outliers and effectiveness of the derail preservation empirically. Secondly, the above clustering algorithms usually make use of the Euclidean distance measure. However, remote sensing images are data with non-Euclidean structure in most cases. Fuzzy clustering algorithms may not function well in these situations [4]. Lastly, the traditional optimization methods mainly belong to mountain-climbing algorithms. Hence, it is easy for them to get stuck in local optima.

To overcome the above problems, in this paper, an adaptive differential evolution fuzzy clustering algorithm with spatial information and kernel metric for remote sensing imagery, namely KADESFC, is proposed. In KADESFC, the trade-off parameter is determined adaptively by introducing the concept of the entropy, which assigns more spatial constraint to pixels with greater entropy, due to the fact that the pixels with larger entropy are much more uncertain. Meanwhile, kernel metric distance is introduced, the priority of which to Euclidean distance is that kernel metric distance can transform the original low-dimensional feature space into a higher dimensional feature space, making the data linearly separable in the high-dimensional feature space. Moreover, differential evolution algorithm (DE) [5] is one of the recently growing areas of research in evolution algorithm and DE has been proved to have powerful global search capability in many applications [1, 6-7]. Hence, DE is used to optimize the proposed objective function in this paper.

The rest of this paper is organized as follows: Section 2 describes the proposed algorithm. Section 3 presents experimental results. Section 4 gives our conclusions.

2 Adaptive Differential Evolution Fuzzy Clustering Algorithm with Spatial Information and Kernel Metric (KADESFC)

In this paper, an adaptive differential evolution fuzzy clustering algorithm with spatial information and kernel metric for remote sensing imagery, namely KADESFC, is

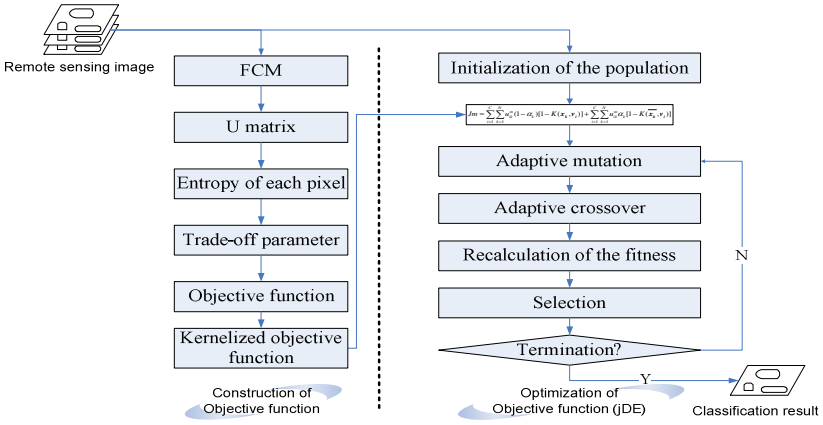


Fig. 1. Flowchart of adaptive differential evolution fuzzy clustering algorithm with spatial information and kernel metric (KADESFC)

proposed. The process of KADESFC contains two process: 1) the construction the objective function; and 2) the optimization of the objective function. The flowchart of KADESFC is as Fig. 1, which will be described in detail in the following part.

2.1 The Objective Function of the Proposed Algorithm

The objective function of the proposed algorithm is as follows.

$$Jm = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m}(1-\alpha_k)[1-K(\boldsymbol{x_k},\boldsymbol{v_i})] + \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m}\alpha_k[1-K(\overline{\boldsymbol{x_k}},\boldsymbol{v_i})] \quad (1)$$

$$U \in \{u_{ik} \in [0,1] \mid \sum_{i=1}^{C} u_{ik} = 1 \; and \; 0 < \sum_{k=1}^{N} u_{ik} < N\} \quad (2)$$

where $\boldsymbol{x_k}$ is a vector representing the *k*-th pixel. *N* is the total number of pixels. $\{\boldsymbol{v_i}\}_{i=1}^{c}$ are the centroids of the clusters. The parameter *m* is a weighting exponent on each fuzzy membership. The array $U=[u_{ik}]$ is a fuzzy membership matrix satisfying (2). $\overline{\boldsymbol{x_k}}$ represents the mean of the pixels falling into a window around $\boldsymbol{x_k}$. $K(\boldsymbol{x_k},\boldsymbol{v_i})$ is the kernel metric distance between $\boldsymbol{x_k}$ and $\boldsymbol{v_i}$ based on Radial Basis Function (RBF) kernel [8]. The trade-off parameter α_k is used to control the effect of the neighbor term, which is determined by the entropy of the *k*-th pixel. The formula of entropy is as follows.

$$E_k = -\sum_{i=1}^{C} u_{ik} \log_2 u_{ik} \quad (3)$$

where E_k is the entropy of the *k*-th pixel. The bigger the entropy of a pixel, the more uncertain the pixel is. Our motivation of introducing the entropy is to put more spatial constraints on the pixels with more uncertainty. In order to get the entropy of each pixel, FCM is applied onto the remote sensing image first and the membership matrix (U matrix) can be acquired. Then, the entropy of each pixel is calculated by using (3). Because the range of the entropy is not 0-1, the entropy need to be increased to 0-1 linearly. The above process is shown in the left part of Fig. 1.

2.2 Optimization of the Objective Function

An adaptive DE [9], jDE, is used to optimize the objective function (1). The whole optimization process is shown in the right part of Fig. 1 and can be implemented according to the following steps.

Step 1. Initialization of the population. Fig. 2 is an individual encoding example. Considering the adaptive mutation and crossover in jDE, the mutation scale factor F and the crosso`ver constant CR need to be encoded into the individual.

9.8	15.6	8.2	17.3	7.5	14.6	0.8	0.3
Cluster center 1		Cluster center 2		Cluster center 3		CR_i	F_i

Fig. 2. the example of individual encoding(eg. three cluster centers and two dimensions)

Step 2. Calculation of the fitness of each individual using (1).

Step 3. Adaptive mutation and crossover. In jDE, the mutation operator amounts to creating a donor vector $\boldsymbol{V}_i(t)=[v_{i,1}(t),v_{i,2}(t),...,v_{i,D}(t)]$ for changing each individual. The mutation process can be expressed as follows.

$$\boldsymbol{V}_i(t)=\boldsymbol{X}_{r_1^i}(t)+F_i(\boldsymbol{X}_{r_2^i}(t)-\boldsymbol{X}_{r_3^i}(t)) \tag{4}$$

where $\boldsymbol{X}_{r_1^i}(t),\boldsymbol{X}_{r_2^i}(t),\boldsymbol{X}_{r_3^i}(t)$ are picked up randomly from the population.

After mutation operator, crossover is operated between the donor vector $\boldsymbol{V}_i(t)$ and the target vector $\boldsymbol{X}_i(t)$, generating a trial vector $\boldsymbol{U}_i(t)=[u_{i,1}(t),u_{i,2}(t),...,u_{i,D}(t)]^T$. The crossover operator can be implemented as follows.

$$u_{i,j}(t)=\begin{cases}v_{i,j}(t), if\ (rand_{i,j}(0,1)\le CR_i\ or\ j=j_{rand})\\ x_{i,j}(t), otherwise\end{cases} \tag{5}$$

In jDE, there are two main parameters including F and CR. The F_i and CR_i can update according to (6) and (7) [9].

$$F_i^{'}=\begin{cases}0.1+0.9\times rand(0,1), & if\ rand(0,1)<0.1\\ F_i, & otherwise\end{cases} \tag{6}$$

$$CR_i^{'}=\begin{cases}rand(0,1), & if rand(0,1)\le 0.1\\ CR_i, & otherwise\end{cases} \tag{7}$$

Step 4. Recalculation of the fitness of the offspring using the objective function (1).

Step 5. Selection. The selection operator is to decide who is the winner between the target vector $\mathbf{X}_i(t)$ and the trial vector $\boldsymbol{U}_i(t)$. The target vector of the next generation can be generated by selection operator as follows:

$$\boldsymbol{X}_i(t+1)=\begin{cases}\boldsymbol{U}_i(t), if\ f(\boldsymbol{U}_i(t))\le f(\boldsymbol{X}_i(t))\\ \boldsymbol{X}_i(t), otherwise\end{cases} \tag{8}$$

Step 6. Elite strategy. In order to speed the coverage of iteration and enhance the efficiency of optimization, elite strategy is applied, which preserves the individual with the best fitness found so far.

Step 7. Terminal condition. Repeat step 3 to step 6 until the terminal condition is met. The terminal condition is to either reach the maximum number of iterations or begin to stagnate for update of the best individual.

Step 8. Clustering image by defuzzying the membership matrix corresponding the best individual. That is, each pixel is assigned to the class with the greatest membership.

3 Experimental Results

3.1 Parameter Setting

The proposed algorithm was compared with several clustering algorithms including FCM, FCM_S1 [3] and DE_KFCM. It should be noted that DE_KFCM used the kernel metric distance and DE without introducing the spatial information. The kernel bandwidth used in DE_KFCM and KADESFC was set as the number of dimension empirically. The size of population was 5×D [10], where D was the dimension of the image. The number of max generation was 100. In order to enhance the efficiency of optimization, we set another parameter gamma=10, meaning that if the best individual cannot update for ten consecutive generations, the iteration will stop.

3.2 Experiment 1-FLC Multi-spectral Image

In experiment 1, a Flightline C1 image of Tippecanoe County, Indiana, US, was used, which was acquired from the M7 scanner, at a resolution of 36.25 m × 36.25 m, a size of 97×102 pixels and twelve bands, in June 1966. This image contains four classes: corn , oat, red clover, and wheat. The original image and ground truth image are shown in Fig. 3(a)-(b).

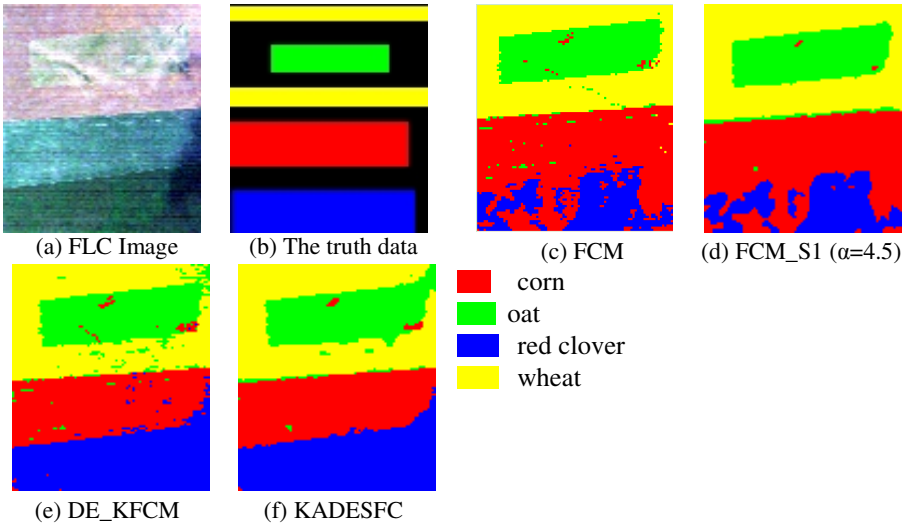


Fig. 3. FLC image and its classification results

Fig. 3(c)-(f) illustrate the classification results using FCM, FCM_S1, DE_KFCM and KADESFC, respectively. Visually, the classification results of FCM and DE_KFCM have many isolated pixels compared with FCM_S1 and KADESFC, due to the disregard of spatial information. DE_KFCM and KADESFC achieve better classification results for red clover class, as many corn pixels are incorrectly classified into red clover class in the results of FCM and FCM_S1. This largely results from the normalization of the orig-

inal image in DE_KFCM and KADESFC and the optimization of differential algorithm. As a whole, KADESFC achieves the best visual accuracy.

To compare the above algorithms quantitatively, overall accuracy (OA) and kappa coefficient are listed in Table 1. KADESFC obtains the best OA, 99.34%, with the gains of 12.25%, 10.04% and 1.70% over FCM, FCM_S1 and DE_KFCM, respectively. The comparison is consistent with the above qualitative finding. Based on the above analysis, KADESFC outperforms three other classifiers.

Table 1. Comparison of classification accuracy for FLC image

	Class	FCM	FCM_S1	DE_KFCM	KADESFC
Producer's accuracy (%)	Corn	98.05	99.93	97.55	99.93
	Oat	97.84	98.77	97.69	97.69
	Red clover	62.67	66.04	99.65	99.93
	Wheat	99.13	99.91	95.24	98.88
AA(%)		89.42	91.16	97.53	99.11
OA(%)		87.09	89.30	97.64	99.34
Kappa		0.8278	0.8539	0.9678	0.9911

3.3 Experiment 2-Wuhan TM Image

In experiment 2, Wuhan TM image is used, which is a 30 m resolution multi-spectral Landsat TM image of Wuhan City, China with the size of 400×400 pixels, and six bands. This region of the image is expected to contain five classes: river, vegetation, lake, bare soil, building. The original Wuhan TM image and ground truth image are shown in Fig. 4 (a)-(b).

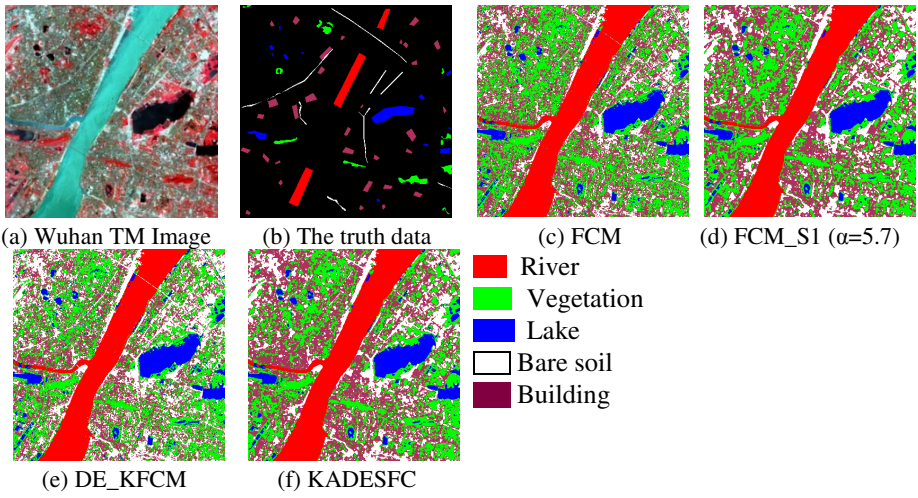


Fig. 4. Wuhan TM image and its classification results

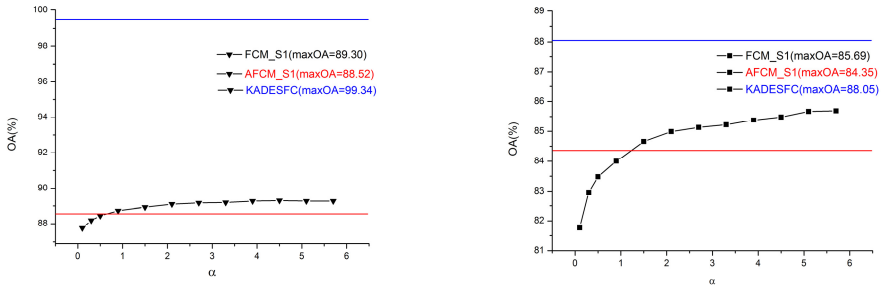
Fig. 4(c)-(f) illustrate the classification results using FCM, FCM_S1, DE_KFCM and KADESFC. For the FCM and DE_FCM method, many isolated pixels exist for the lack of the spatial information. For the FCM method, the vegetation and building classes are classified incorrectly and the bare soil class is misclassified into two classes. FCM_S1 and KADESFC give better visual results compared with FCM and DE_KFCM with more unwanted isolated pixels. KADESFC achieves the best visual accuracy.

To compare the algorithms quantitatively, overall accuracy (OA) and kappa coefficient are also listed in Table 2. KADESFC obtains the best OA, 88.05%, with the gains of 6.39%, 2.36% and 3.40% over FCM, FCM_S1 and DE_KFCM, respectively. The quantitative comparison of the four algorithms is consistent with the above qualitative finding. Based on the above analysis, KADESFC outperforms three other classifiers.

Table 2. Comparison of classification accuracy for Wuhan TM image

	Class	FCM	FCM_S1	DE_KFCM	KADESFC
Producer's accuracy (%)	River	100.00	100.00	100.00	100.00
	Vegetation	82.50	89.18	84.13	90.37
	Lake	99.80	100.00	99.88	99.94
	Bare soil	71.80	67.93	66.22	71.17
	Building	61.20	71.58	80.53	78.33
	AA(%)	83.06	85.74	86.15	87.96
	OA(%)	81.66	85.69	84.65	88.05
	Kappa	0.7619	0.8154	0.8036	0.8461

In order to further test the validity of the proposed adaptive trade-off parameter, Fig. 5 shows the OA of FCM_S1 corresponding to different α for the two images. It should be noted that the proposed AFCM_S1 method only uses the adaptive trade-off parameter without introducing the kernel metric distance and differential algorithm. As shown in Fig. 5, although the OA of AFCM_S1 is a little lower than the best OA of FCM_S1, the trade-off parameter of AFCM_S1 is determined adaptively. In addition, the OA of KADESFC is higher than the other methods significantly.



(a) FLC image (b) Wuhan TM image

Fig. 5. The influence of α for FCM_S1

4 Conclusion

This paper proposed an adaptive differential evolution fuzzy clustering algorithm with spatial information and kernel metric for remote sensing imagery, namely KADESFC. The KADESFC defined an objective function to determine the trade-off parameter adaptively. The kernel metric distance and DE are introduced to further enhance the clustering performance. The experiment results show that the KADESFC consistently outperforms other clustering algorithms visually and quantitatively. Hence, it provides an effective method for remote sensing image clustering.

References

1. Zhong, Y., Zhang, S., Zhang, L.: Automatic Fuzzy Clustering Based on Adaptive Multi-Objective Differential Evolution for Remote Sensing Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2013), doi:10.1109/JSTARS.2013.2240655
2. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Transactions on Medical Imaging 21(3), 193–199 (2002)
3. Chen, S., Zhang, D.: Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 34(4), 1907–1916 (2004)
4. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern Recognition 41(1), 176–190 (2008)
5. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
6. Zhong, Y., Zhang, L.: Remote Sensing Image Subpixel Mapping Based on Adaptive Differential Evolution. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 42(5), 1306–1329 (2012)
7. Hejazi, H., Mohabati, H., Hosseinian, S., Abedi, M.: Differential evolution algorithm for security-constrained energy and reserve optimization considering credible contingencies. IEEE Transactions on Power Systems 26(3), 1145–1155 (2011)
8. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1999)
9. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation 10(6), 646–657 (2006)
10. Gämperle, R., Müller, S.D., Koumoutsakos, P.: A parameter study for differential evolution. Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation 10, 293–298 (2002)

Dynamic EM in Neologism Evolution

Martin Emms

Dept. of Computer Science, Trinity College, Dublin, Ireland
www.scss.tcd.ie/Martin.Emms

Abstract. Research on unsupervised word sense discrimination typically ignores a notable dynamic aspect, whereby the prevalence of a word sense varies over time, to the point that a given word (such as 'tweet') can acquire a new usage alongside a pre-existing one (such as 'a Twitter post' alongside 'a bird noise'). This work applies unsupervised methods to text collections within which such *neologisms* can reasonably be expected to occur. We propose a probabilistic model which conditions words on senses, and senses on times and an EM method to learn the parameters of the model using data from which sense labels have been deleted. This is contrasted with a static model with no time dependency. We show qualitatively that the learned and the observed time-dependent sense distributions resemble each other closely, and quantitatively that the learned dynamic model achieves a higher tagging accuracy (82.4%) than the learned static model does (76.1%).

Keywords: neologism sense EM.

1 Introduction

Language is subject to continuous change, one aspect of which is lexical: an existing word can acquire a new usage. Arguably content that has been relatively recently authored for access via the internet is relatively rich in such *neologisms*.

For example, a post to *Twitter* is known as a *tweet*. This is a new usage, predated by the use of *tweet* to refer to a particular kind of bird noise. Related to this *Twitter*-related noun usage of *tweet* there is also *tweeted* and *tweeter*. Before the era of Twitter, the pre-eminent usage of *tweeter* was to designate a particular piece of audio hardware.

The n-gram *totes* has acquired a usage where it substitutes for *totally*, (as in *I was totes steamed*). This is pre-dated by several other usages, one where it is the plural of *tote*, itself ambiguous between designating a certain kind of bag or designating a man who takes bets at a race course.

The n-gram *bricked* has acquired a usage meaning roughly to render a piece of equipment, often a phone, entirely unresponsive. This is pre-dated by a usage referring to a construction process involving bricks.

Whilst there are many interesting questions as to the mechanisms at work in creating these neologisms, this article is concerned with the simple fact that such new usages add to the ambiguity-resolution problem that is a feature of many NLP tasks, such as information retrieval or machine translation.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 286–293, 2013.

To illustrate the fact that some of these neologisms are not well handled, we sought their translation into German via Google Translate[1]:

English	German (via Google Translate)
he is a regular tweeter	*er ist ein regelmaessiger Hochtoener*
he has bricked my phone	*er hat mein Handy zugemauert*

The chosen translation of *tweeter*, namely *Hochtoener*, would have been appropriate to its audio hardware sense. It is not appropriate to its Twitter-related sense. Similarly the chosen translation of *bricked*, namely *zugemauert*, would have been appropriate to a construction setting, but is not appropriate to the context of rendering a phone inert.[2]

The terms *unsupervised sense disambiguation* or *word sense discrimination* are often used to describe a process which takes as input a corpus of examples of a word's use – unlabelled with any sense information – and returns a partitioning of the examples into groups, so that all members of a group exhibit a particular sense or usage [9,7,5,2]. Often in such systems, each cell of the partition is associated with parameters, and via these for any given example of the word's use, a number can be derived which can be interpreted as strength of membership in that cell. For example, in Schütze's work [9], each cell of the partitition is associated with a centroid vector, and distance from this indicates strength of membership. The outputs of such word sense discrimination systems can have practical value. For example, for a probabilistically trained MT system, one might be able to then recognise that in a sentence to be translated, a given word belongs only weakly to any of the cells of the partition for that word in the training corpus for the system and so is likely to be poorly translated [11]. There is also work reporting improved performance in information retrieval [10].

The work to be presented below is in the area of word sense discrimination, but its innovative aspect is that we seek to address the above-noted fact that language is a moving target. One can expect that the prevalence of use of a particular sense varies with time. For the kind of neologisms mentioned above – new usages of an existing word – it should be the case that prior to a particular point in time, usage with the new sense is non-existent. For example, Twitter was founded in 2006, so that the Twitter-related sense of *tweet* should be absent from text authored prior, or substantially prior, to that date.

Thus our corpus of word occurrences will span a stretch of time. If there is change in the prevalence of use of a particular sense over that time period, possibly even the emergence of a particular sense at a particular time, the question we wish to address is whether that can be detected by an unsupervised method, and whether one particular model is better than another to accomplish this.

We will propose a probabilistic model of word use which conditions words on senses, and senses on times, and we will propose a *dynamic EM* method for

[1] Executed May 2013.

[2] Some internet search and consultation with German-speaking colleagues suggest that *gebrickt* would be the German equivalent.

estimating the parameters of the model from data in which times and words are visible, but senses hidden. This will be contrasted with a static variant with senses unconditioned on times. We will give qualitative evidence that the method is able to infer changes in the prevalence of a sense over time, and quantitative evidence that the dynamic variant out performs the static variant, achieving an accuracy of 82.4% vs 76.1%, when the partitions are mapped to manual assigned senses.

2 Dynamic and Static Models

The following snippet of text, dating from 2001, is an example of the construction-related sense of *bricked*:

... In 1611 she was **bricked** *into one of the rooms ...*

and the next snippet, dating from 2011, illustrates the 'render inert' sense:

I've tried to flash a custom ROM and now I think I've **bricked** *my phone*

Where T is an occurrence of a target ambigous word, let $\boldsymbol{W}$ be a sequence whose first l elements are the l words to the left of T and whose last r elements are the r words to the right of T. Assume the document in which T features was authored in a particular year, represented by variable Y. Let S range over available senses for target ambiguous term. We assume that the occurrence T uses a particular sense. If we suppose there are three available senses, with $S = 1$ for the construction-related sense and $S = 2$ for the phone-related sesnse, then choosing $l = r = 5$, the two examples above might be represented thus:[3]

$$Y = 2001,\ S = 1,\ \boldsymbol{W} = \langle L, In, 1611, she, was, into, one, of, the, rooms\rangle$$
$$Y = 2011,\ S = 2,\ \boldsymbol{W} = \langle and, now, I, think, I've, my, phone, R, R, R\rangle$$

We wish to consider possible probability models for combinations of the variables Y, S and $\boldsymbol{W}$. Without loss of generality, using the chain rule, we have

$$p(Y, S, \boldsymbol{W}) = p(Y) \times p(S|Y) \times p(\boldsymbol{W}|S, Y)$$

Considering the last term in the above, we will assume that $\boldsymbol{W}$ is conditionally independent of Y given S, that is, $p(\boldsymbol{W}|S, Y) = p(\boldsymbol{W}|S)$, so that

$$p(Y, S, \boldsymbol{W}) = p(Y) \times p(S|Y) \times p(\boldsymbol{W}|S) \quad (1)$$

Whilst this is a simplifying assumption, there is some plausibility to the idea that given the concept that is being conveyed, there is some substantially time-independent aspect of the expected accompanying vocabulary. The assumption drastically reduces the number of parameters that need to be estimated: with

[3] The context is padded with L and R if its not wide enough.

a 10-year time span and a 3-way sense ambiguity, the word probabilities are conditioned on 3 settings rather than 30. We further treat $p(\boldsymbol{W}|S)$ as if $\boldsymbol{W}$ represents $|\boldsymbol{W}|$ trials of a single multinomial variable, whose values are the different possible words, hence $p(\boldsymbol{W}|S) = \prod_i (p(\boldsymbol{W}_i|S)$ (essentially a so-called unigram language model [4]).

The second term in (1) directly expresses the idea that the prevalence of a given sense can vary with the year. For example, on a-priori grounds, we can be sure that the Twitter-relatd sense of *tweet* has zero probability for $Y = 1972$.

Whilst (1) gives a *dynamic* model, the further simplifying assumption that the sense is *independent* of the time, $p(S|Y) = p(S)$, gives the *static* model (2)

$$p(Y, S, \boldsymbol{W}) = p(Y) \times p(S) \times p(\boldsymbol{W}|S) \tag{2}$$

If all mentions of the time Y are removed from the data and (2) then the static model is essentially the probabilistic model considered in several works ([3,5]).

If the probabilities mentioned on the righthand sides of the equations (1) and (2) are known then for an instance with given values for the Y and $\boldsymbol{W}$, the instance can be classified (resp. assigned a probability) by choosing the value in $val(S)$ that maximises (resp. taking the sum over all values in $val(S)$ of) the joint probability $p(Y = y, S, \boldsymbol{W} = \boldsymbol{w})$.[4]

In our training data, the sense variable S will be *hidden*, and to estimate the parameters of the proposed model from this data we propose to use an Expectation-Maximisation (EM) approach to obtain values for the parameters – for what is essentially the *static* model of (2), this is discussed in [3,5]. The development below is based on the idea of repeatedly calculating expected completions of incomplete data, and then deriving new parameters from these by maximum likelihood estimation. [6] is a good exposition of this perspective on EM.

Let $\mathcal{T} = \ldots (Y^d, \boldsymbol{W}^d) \ldots$, be a corpus of instances: note the sense S is not present. Then in outline the training algorithm consists in iterations of the following pair of steps, mapping the estimates $\theta_n(p(Y), p(S|Y), p(\boldsymbol{W}|S))$ to new estimates $\theta_{n+1}(p(Y), p(S|Y), p(\boldsymbol{W}|S))$.

(Exp) *generate a virtual corpus of disambiguated instances by treating each training instance* $(Y^d, \boldsymbol{W}^d)$ *as standing for* all *possible completions with a sense,* $(Y^d, S, \boldsymbol{W}^d)$*, weighting* each *by its conditional probability* $P(S|Y^d, \boldsymbol{W}^d)$*, under current probalities* $\theta_n(p(Y), p(S|Y), p(\boldsymbol{W}|S))$

(Max) *apply maximum likelihood estimation to the virtual corpus to derive new estimates* $\theta_{n+1}(p(Y), p(S|Y), p(\boldsymbol{W}|S))$.

Concretely, where D is the size of the training data, let γ be a table of size $D \times |val(S)|$ to store for each data point d the conditional probabilities $P(S = s|Y = y^d, \boldsymbol{W} = \boldsymbol{w}^d)$ for each $s \in val(S)$. For each d, $\gamma[d][s]$ is calculated according to

[4] For any variable X with discrete values, let $val(X)$ be its set of possible values.

$$\gamma[d][s] = \frac{P(Y = y^d, S = s, \boldsymbol{W} = \boldsymbol{w}^d)}{\sum_{s' \in val(S)} P(Y = y^d, S = s', \boldsymbol{W} = \boldsymbol{w}^d)}$$

Supposing γ has been filled by a traversal of the training data $\{..., (Y^d, \boldsymbol{W}^d), ...\}$. For the dynamic model of (1) the re-estimation formale for the (Max) step are as follows

For each $y \in val(Y)$, and $s \in val(S)$

$$P(S = s|Y = y) = \frac{\sum_d (\text{if } Y^d = y \text{ then } \gamma[d][s] \text{ else } 0)}{\sum_d (\text{if } Y^d = y \text{ then } 1 \text{ else } 0)}$$

For each $s \in val(S)$

$$P(w|S = s) = \frac{\sum_d (\gamma[d][s] \times freq(w \in \boldsymbol{W}^d))}{\sum_d (length(\boldsymbol{W}^d))}$$

For the static model of (2), the re-estimation of $P(S)$ will be via

$$P(S = s) = \frac{\sum_d (\gamma[d][s])}{D}$$

3 Experiments

As mentioned earlier *bricked* seems to have acquired a sense relating to rendering a technical device (often a phone) inert, in addition to others senses, principally a building-related sense. To test the approach for *bricked*, a corpus of time-stamped occurrences was required. This was obtained using a facility that Google has offered for some time in which it is possible to specify a time period for searched documents. Eleven year-long search periods were used, (1/1/2001-31/12/2002) ... (1/1/2011-31/12/2012).

For each of these time periods, the search item was set to *bricked* and the first 20 hits retained. These were then used to give time specific occurrences of the search item, and a window of 5 words to left and right was used to define $\boldsymbol{W}$, giving 220 occurrences, 20 for each year. For later evaluation purposes these were manually inspected and labelled with one of three sense tags: *WALLS*, for the construction-related sense, *PHONE* for the render-inert sense, and *OTHER* of any other sense. Two examples of this data were given at the beginning of Sect. 2.

Based on the manually assigned sense labels, Fig. 1 shows for each year the relative proportions of occurences for the WALLS and PHONE senses of *bricked*. It is clear from this plot that, at least for the sample corpus obtained, there is a substantial shift in the proportion of occurrences accounted for by *WALLS* and *PHONE* senses. In 2001, *WALLS* predominates, and *PHONE* is negligible, whilst by 2011 the situation is reversed, and *PHONE* predominates and *WALLS*

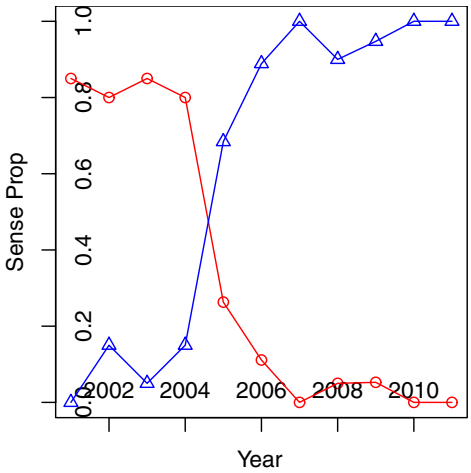


Fig. 1. Observed per-year proportions of two senses of *bricked* from 2001 to 2011 (*WALLS* shown as circles, *PHONE* shown as triangles)

is negligible, with the cross-over happening somewhere between the beginning of 2004 and the end of 2005. Combining all years together, the relative proportions are *WALLS* (0.357) and *PHONE* (0.6)

We ran the EM training algorithm on this data set, assuming the dynamic model of (1). Recall that in this training the sense values are *hidden*. When the sense variable is summed-out, the EM training increased the likelihood of the data, as it theoretically should. Of more interest though are the conditional sense-given-year probabilities after the training process has converged and these are shown in Fig. 2. Qualitatively one can see that the obtained values resemble quite closely the empirical sense distributions which were shown in Fig. 1.

Recall that although the unsupervised EM training operated on an unlabelled version of the data, there are manually assigned labels available for the data. If one defines a mapping, σ, from the sense identifiers of the model, $\{1, 2, 3\}$, to the sense identifiers in the data, $\{WALLS, PHONE, OTHER\}$, one can evaluate the accuracy of the labelling produced by the model at the end of the unsupervised EM training, modulo the mapping σ. As there are many possible mappings, following the methodology of [7], we report the highest accuracy under all mappings. The accuracy obtained in this way is 82.4%. When instead the *static* model is used from (2), the trained model's accuracy is somewhat lower: 76.1%.

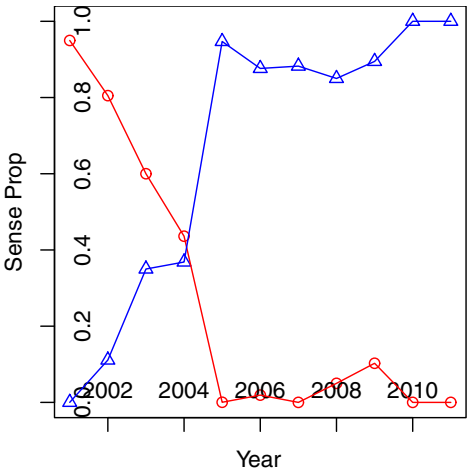


Fig. 2. Values of $P(S|Y)$ across the years after EM training using the dynamic model

4 Discussion and Conclusions

We have given some preliminary evidence concerning the possibility to use an EM method to infer senses on data where the sense distributions are intuitively time dependent. A model incorporating this time dependence was shown to outperform a static model. As far as we are aware, this has not been done before.

Work on *Dynamic Topic Models* [1] takes a somewhat analogous approach, with *topic* (such as 'Atomic Physics' or 'Neuroscience') as a hidden, time-dependent variable, and probabilities on a *document*'s words which are topic and time dependent. Besides being applied to document topics rather than word senses, the method used – variational approximation – differs substantially to that used here. It remains for future work to make a more detailed comparison. A preliminary observation would be that the additional *time*-dependency of word probabilities for topics seems more natural than it is for word-senses.

Although the current work is concerned with an interaction between senses and times, it still treats senses as somehow *eternal*: abstractly for a given span of time and a given word, there is a fixed set of available senses and what varies is the probability for a sense over time. [8] reports work in historical linguistics on tracing semantic change which uses the context vectors of [9] to analyse changes in word senses themselves. For example, in comparison to hundreds of years ago, *dog* (resp. *deer*) has come to have a much broader (resp. narrower) sense, and they claim to be able to map this to changes in the geometry of the set of vectors for a word over time. It remains for future work to consider more closely the relation between the work of [8] and the problem addressed here.

This work was a preliminary investigation. One issue needing attention is the use of Google's time-specific search facility. For the lexical item looked at, the time-stamping seemed seldom glaringly inaccurate, but for others that were

considered, it seemed so. Also, although we made the simplifying assumption that the vocabulary probabilities were independent of the time, given the sense, it is easy to imagine situations where this is not going to be true. One direction for future work would be to investigate outcomes in which this simplifying assumption is not made, but the increased sophistication of such a model certainly suggests that it would demand a far larger corpus.

Besides attempting to use the inferred model to assign sense-group labels to words, one could instead use the approach to attempt to simply recognise that over a particular span of time there appears to be a newly emerging sense, or neologism.

Acknowledgements. This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

References

1. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Cohen, W., Moore, A. (eds.) ICML 2006: Proceedings of the 23rd International Conference on Machine Learning, pp. 113–120. ACM Press, New York (2006)
2. Maldonado-Guerra, A., Emms, M.: First-order and second-order context representations: geometrical considerations and performance in word-sense disambiguation and discrimination. In: Dister, A., Dominique Longrée, G.P. (eds.) Proceeding of JADT 11th International Conference on the Statistical Analysis of Textual Data, pp. 676–686. LASLA (2012)
3. Manning, C., Schütze, H.: Word Sense Disambiguation. In: Foundations of Statistical Language Processing, 6th edn., pp. 229–264. MIT Press (2003)
4. Manning, C.D., Raghavan, P., Schütze, H.: Language models for information retrieval. In: Introduction to Information Retrieval. Cambridge University Press (2009)
5. de Marneffe, M.C., Dupont, P.: Comparative study of statistical word sense discrimination. In: Purnelle, G., Fairon, C., Dister, A. (eds.) Proceedings of JADT 2004 7th International Conference on the Statistical Analysis of Textual Data, pp. 270–281. UCL Presses Universitaire de Louvain (2004)
6. Prescher, D.: A tutorial on the expectation-maximization algorithm including maximum-likelihood estimation and em training of probabilistic context-free grammars. Computing Research Repository (2004)
7. Purandare, A., Pedersen, T.: Word sense discrimination by clustering contexts in vector and similarity spaces. In: Ng, H.T., Riloff, E. (eds.) Proceedings of CoNLL 2004, Boston, MA, USA, pp. 41–48 (2004)
8. Sagi, E., Kaufmann, S., Clark, B.: Tracing semantic change with latent semantic analysis. In: Allan, K., Robinson, J.A. (eds.) Current Methods in Historical Semantics, pp. 161–183. Mouton de Gruyter, Berlin (2012)
9. Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–123 (1998)
10. Véronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech and Language 18(3), 223–252 (2004)
11. Vickrey, D., Biewald, L., Teyssier, M., Koller, D.: Word-sense disambiguation for machine translation. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 771–778. Association for Computational Linguistics, Stroudsburg (2005)

Estimation of the Regularisation Parameter in Huber-MRF for Image Resolution Enhancement

Sakinah Ali Pitchay[1] and Ata Kabán[2]

[1] Faculty of Science and Technology, Universiti Sains Islam Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia
[2] School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
{S.A.Pitchay,A.Kaban}@cs.bham.ac.uk, sakinah.ali@usim.edu.my

Abstract. The Huber Markov Random Field (H-MRF) has been proposed for image resolution enhancement as a preferable alternative to Gaussian Random Markov Fields (G-MRF) for its ability to preserve discontinuities in the image. However, its performance relies on a good choice of a regularisation parameter. While automating this choice has been successfully tackled for G-MRF, the more sophisticated form of H-MRF makes this problem less straightforward. In this paper we develop an approximate solution to this problem, by upper-bounding the partition function of the H-MRF. We demonstrate the working and flexibility of our approach in image super-resolution experiments.

Keywords: Huber prior, Hyper-parameter Optimisation, Super-resolution.

1 Introduction

Image super-resolution seeks to enhance the resolution of an image beyond of that of the detector arrays of the cameras, by appropriately combining together the available low resolution measurements. The difficulty lies with the ill-posed nature of this problem, which often requires the solution of a severely under-determined system. The use of an appropriate image model, or prior to regularise the problem is therefore crucial. To this end, Gaussian Markov Random Fields (G-MRF) represents a convenient common choice which, however is known to have a tendency to blur the edges of the super-resolved image [4,7]. Huber-Markov Random Fields (H-MRF) have been designed to address this problem, and represent a state of the art approach for super-resolution [4,7]. The Huber function has longer tails than the Gaussian, hence the edges of the image get less penalised. In addition, the Huber prior is convex; hence, if coupled with a Gaussian observation model, it guarantees a unique solution.

However, the good performance of H-MRF relies on a good choice of its free parameters. While automating this choice has been successfully tackled for G-MRF [2], the same is currently not available for H-MRF. Due to the intractable form of the partition function of Huber-MRF, a principled model-based estimation

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 294–301, 2013.

of the hyper-parameters has been found problematic [7], and various heuristic approaches have been in use [4,7] to sidestep the computational burden of cross-validation. In this paper, we develop an approximation, which allows us to obtain an estimate of the regularisation parameter of the H-MRF prior simultaneously with obtaining the resolution enhancement.

2 Model

2.1 The Observation Model

We denote the original, high-resolution image by the column vector $\boldsymbol{z}$ throughout the paper. This is a vectorised form of the image, and contains N pixel intensities. The degradation process will be taken as a linear transform that consists of shifts, blur and down-sampling. With k low resolution versions of the original image, W will denote the k linear transforms stacked into a single $M \times N$ matrix, where M is the total number of observed low resolution pixels. Likewise, the M-dimensional column vector $\boldsymbol{y}$ will contain the vectorised low resolution observations stacked together. This allows us to write the observation model in a simple form, as follows:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\eta} \tag{1}$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is additive noise.

2.2 The Huber-MRF Prior

Huber Markov Random Field (H-MRF) image-prior is studied in [4,5,7] and the univariate Huber density is defined as:

$$p(u|\alpha) = \frac{1}{Z(\alpha)} \exp\{-\alpha g(u)\} \tag{2}$$

where

$$g(u|T) = \begin{cases} u^2, & \text{if } |u| < T; \\ 2T|u| - T^2, & \text{if } |u| \geq T. \end{cases} \tag{3}$$

is the Huber function, T is a threshold parameter, and $Z(\alpha)$ is the partition function, independent of u. The Huber function takes a threshold parameter T that specifies the value at which it diverts from being quadratic to being linear. If the threshold T is large, then the H-MRF prior reduces to a Gaussian-MRF image-prior. On the other hand, if $T \to 0$, the Huber prior is equivalent to an L1-regularisation based prior.

The first order MRF model posits that each pixel intensity depends only on its immediate neighbours. Here we condition on the average of the four cardinal neighbours, following [3,4], which appears to be a both simple and efficient choice. For convenient notation, the $N \times N$ matrix $\boldsymbol{D}$ is defined with values 1 in its diagonal entries, d_{ij} = -1/4 whenever j is a cardinal neighbour of i, and 0

everywhere else. $\boldsymbol{D}_i$ will denote the i-th row of this matrix. Then we can write the difference between the i-th pixel and its average neighbours simply as $\boldsymbol{D}_i\boldsymbol{z}$. With these notations in place, the H-MRF prior is the following:

$$p(\boldsymbol{z}|\alpha) = \frac{1}{Z_{MRF}(\alpha)} \exp\left\{-\alpha \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})\right\} \tag{4}$$

The partition function of the MRF, $Z_{MRF}(\alpha)$, is independent of $\boldsymbol{z}$.

3 MAP Estimation of $\boldsymbol{z}$

The most probable high resolution image is given by maximising the log posterior of $\boldsymbol{z}$, that is, up to additive constants, equivalent to minimising the negative log of the joint likelihood:

$$L(\boldsymbol{z}) = -\log p(\boldsymbol{y}|\boldsymbol{W}\boldsymbol{z})p(\boldsymbol{z}) \tag{5}$$

$$= \frac{1}{2\sigma^2}(y - Wz)^2 + \alpha \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z}) \tag{6}$$

Notice the normalising term $Z(\alpha)$ is constant here, hence it is not needed. This optimisation is then fairly straightforward. Being a convex objective, any non-linear optimiser will guarantee a global optimum. The gradient of the prior term is:

$$\frac{\partial - \log p(\boldsymbol{z}|\alpha)}{\partial \boldsymbol{z}} = \alpha \sum_{i=1}^{N} g'(\boldsymbol{D}_i\boldsymbol{z}) = \alpha \boldsymbol{D} g'(\boldsymbol{D}\boldsymbol{z}) \tag{7}$$

where the derivative function g'() acts component-wise, and $\boldsymbol{D}$ is the $N \times N$ matrix with rows $\boldsymbol{D}_i$. We employed this in a conjugate gradient optimisation procedure.

The high-resolution estimate is then given by the solution of the stationary system of equations:

$$\boldsymbol{W}^T(\boldsymbol{y} - \boldsymbol{W}\boldsymbol{z}) + \lambda \boldsymbol{D} g'(\boldsymbol{D}\boldsymbol{z}) = 0 \tag{8}$$

where both α and the variance of the likelihood term are now absorbed in the the parameter $\lambda := \alpha\sigma^2$ may be referred to as the regularisation parameter.

4 Determining $\boldsymbol{\lambda}$ by Integrating over $\boldsymbol{\alpha}$

To deal with the hyper-parameter α, we may choose to place the Jeffreys prior (termed often as the ignorance prior) on it, and integrate α out from the model. This approach is the so-called *Bayesian regularisation*. It was first introduced in [8], and subsequently shown to work well for a number of models in the literature. In our case, the Jeffreys prior is:

$$p(\alpha) \propto \frac{1}{\alpha} \tag{9}$$

Hence, the marginal prior becomes:

$$p(\boldsymbol{z}) = \int_0^\infty p(\boldsymbol{z}|\alpha)p(\alpha)d\alpha \tag{10}$$

where the conditional prior $p(\boldsymbol{z}|\alpha)$ is the one defined in eq.(4). However, since we have an MRF model, the partition function $Z_{MRF}(\alpha)$ of our conditional prior $p(\boldsymbol{z}|\alpha)$ is an intractable multivariate integral.

$$Z_{MRF}(\alpha) = \int dz \exp\left\{-\alpha \sum_{i=1}^{N} g(\boldsymbol{D}_i \boldsymbol{z})\right\} \tag{11}$$

It does not depend on $\boldsymbol{z}$, but it does depend on α, — hence we cannot ignore it when we try to estimate both α and $\boldsymbol{z}$ simultaneously. To approximate this integral, we proceed in two steps: we create a reasonably tight upper-bound to the partition function of the univariate Huber density; then we use this in a pseudolikelihood approximation of the MRF partition function.

4.1 Upper-Bounding Z(α)

$Z(\alpha)$ may be written with the use of the error function $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$, it is however uneasy to work with this directly. Instead, we create a convenient analytic upper-bound as the following:

$$Z(\alpha) = \int \exp(-\alpha g(u))du = \frac{\exp(-\alpha T^2)}{\alpha T} + \mathrm{erf}(T\sqrt{\alpha})\sqrt{\pi/\alpha} \tag{12}$$

$$\leq \frac{\exp(-\alpha T^2)}{\alpha T} + \frac{\exp(\alpha T^2) - \exp(-\alpha T^2)}{\alpha T} \tag{13}$$

$$= \exp(\alpha T^2)/(\alpha T) \tag{14}$$

Note that, since the upper-bound approximation to $Z(\alpha)$ was made on the interval $[-T, T]$, the bound will be tighter for smaller values of T . Fortunately, this is not restrictive in practice, since it was empirically observed [7] that the sensible range of values is somewhere in [0.01, 0.1]. In this range our bound is tight, as shown in Figure 1. The bound also tightens for smaller α.

4.2 Approximating $\boldsymbol{Z_{MRF}(\alpha)}$

To be able to employ the upper bound derived in the previous section, we shall combine it with the well-known pseudolikelihood approximation on $Z_{MRF}(\alpha)$. Besides its simplicity, the pseudo-likelihood is known to enjoy consistency [1]. It consists of taking each $\boldsymbol{D}_i\boldsymbol{z}$ as if it was independent of $\boldsymbol{D}_j\boldsymbol{z}$, for all $j \neq i$, to make the problem tractable. This allows us to employ our upper-bound derived earlier, for each variable in turn. Thus, we have the following for (11):

$$Z_{MRF}(\alpha) \approx C.\exp\{T^2\alpha\}^N \alpha^{-N} \tag{15}$$

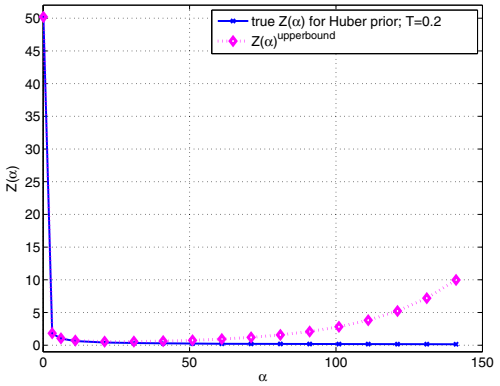


Fig. 1. Tightness of the upper-bound approximation to $Z(\alpha)$, with threshold $T = 0.2$. For smaller values of the threshold, the upper-bound gets practically indistinguishable from the true partition function.

where we used the upper bounds (14) on each $u_i = \boldsymbol{D}_i\boldsymbol{z}$, and C is a constant that subsumes the factors that do not depend on α. Having derived a workable approximation to $Z_{MRF}(\alpha)$, we replace this into (10), so the marginalisation over α becomes tractable, and yields the marginal prior:

$$p(\boldsymbol{z}) \approx C. \int \alpha^{N-1} \exp\left\{-\alpha\left[T^2N + \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})\right]\right\} d\alpha$$

$$\propto \frac{\Gamma(N)}{\left\{T^2N + \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})\right\}^N} \propto \left\{T^2N + \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})\right\}^{-N} \tag{16}$$

where we used that $\int_0^\infty \alpha^{\nu-1} e^{m\alpha} d\alpha = \frac{\Gamma(\nu)}{m^\nu}$.

In consequence, the negative log of the marginal prior is:

$$-\log p(\boldsymbol{z}) = N \log\left\{NT^2 + \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})\right\} + const. \tag{17}$$

4.3 Determining $\boldsymbol{\lambda}$

Now, α is no longer in the model, so we may proceed to optimising z, as before, but now using the marginal prior in place of the conditional prior. Taking the gradient from (17) wrt. $\boldsymbol{z}$,

$$\frac{\partial - \log p(\boldsymbol{z})}{\partial \boldsymbol{z}} = N \frac{1}{NT^2 + \sum_{i=1}^{N} g(\boldsymbol{D}_i\boldsymbol{z})} \boldsymbol{D} g'(\boldsymbol{D}\boldsymbol{z}) \tag{18}$$

we observe this has a similar form as the gradient we had before from the conditional prior, i.e. eq. (7), except that in place of α now we can isolate the following

expression that we may consider an estimate of it:

$$\hat{\alpha} = N \frac{1}{NT^2 + \sum_{i=1}^{N} g(\boldsymbol{D}_i \boldsymbol{z})} \tag{19}$$

Finally, the noise variance is estimated as the following. To avoid singularities, here we use an improper prior $\sigma^2 \propto \exp(-\gamma/(2\sigma^2))$ to discourage zero values, which gives the estimate:

$$\hat{\sigma^2} = \frac{1}{M} \left(\sum_{i=1}^{M} (\boldsymbol{y}_i - \boldsymbol{W}_i \hat{\boldsymbol{z}})^2 + \gamma \right) \tag{20}$$

where M is the total number of pixels in $\boldsymbol{y}$. A small value $\gamma = 10^{-12}$ was used in the experiments, to keep the bias minimal.

Putting together (19) and (20), we obtain the following estimate of the regularisation parameter λ, for use within the stationary equation (7):

$$\hat{\lambda} = \frac{\frac{1}{M} \left(\sum_{i=1}^{M} (\boldsymbol{y}_i - \boldsymbol{W}_i \hat{\boldsymbol{z}})^2 + \gamma \right)}{T^2 + \frac{1}{N} \sum_{i=1}^{N} g(\boldsymbol{D}_i \hat{\boldsymbol{z}})} \tag{21}$$

This is our estimate of the regularisation parameter.

Observe that, equation (21) is quite interpretable: $\hat{\lambda}$ is proportional to the noise variance, and inversely proportional to the sum of the squared threshold plus the average smoothness penalty. The threshold parameter T may be understood as controlling the high frequency vs. low frequency components, which can be set within a sensible range e.g. using synthetic data as proposed in [5].

5 Experiments

We test our method in severely under-determined systems in the first instance. The original image is the 130 × 130 "Butterfly". In each experiment, we compose $\boldsymbol{W}$ from a randomly parameterised homography transform, blur with point spread function (PSF) of standard deviation 0.4, and a specified down-sampling ratio, and then we fix it in order to assess the recovery abilities of the prior. We create three low resolution (LR) versions of the original ground truth, each of size 35 × 35 pixels, and we add a small amount of noise to these ($\sigma = 2.55 \times 10^{-4}$ grey levels). That is, from $M = 1225$ observed LR pixels we try to recover $N = 16900$ (i.e. c.c.a. 4.5 times more) HR ones. Figure 2 shows the results. The averaged bicubic interpolation is given as the strawmans baseline. However, we see that our adaptive method of estimating λ is preferable to using a fixed λ in H-MRF, despite the value $\lambda = 0.01$ should be a sensible one, based on the results in [7]. Our estimate of λ is of the order of 10^{-12}.

In the above experiments, the threshold parameter $T = 0.04$ was used, cf. [6]. To give quantitative figures, and to see how the performance varies with the threshold T , the left hand plot of Figure 3 shows the mean square error (MSE)

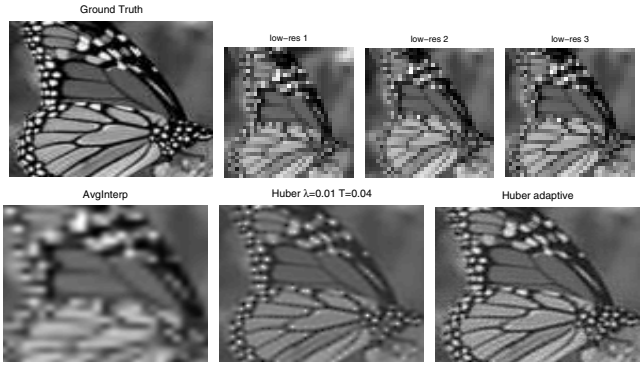


Fig. 2. Top row: The ground truth (130 × 130) of a butterfly and the 3 low-resolution (35 × 35) images; Bottom row: Recovered HR images (130 × 130) comparatively by averaged bicubic interpolation, Huber with fixed $\lambda = 0.01$; and Adaptive Huber.

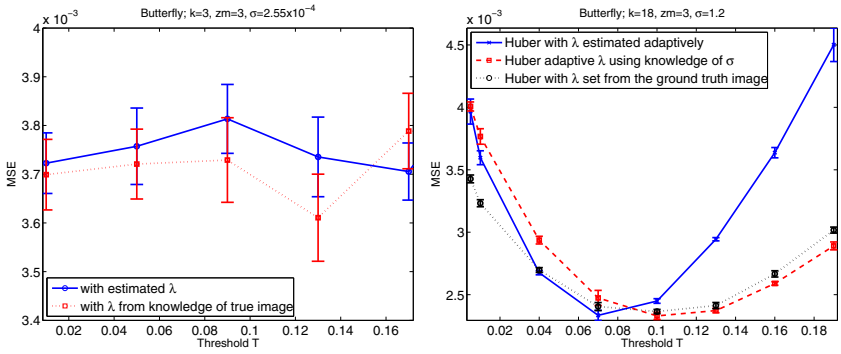


Fig. 3. Dependence on the threshold parameter T and comparison with informed choices of λ. Left: Under-determined system, small noise (k=3 LR images of 35×35 pixels each, additive noise with $\sigma = 2.55\times10^{-4}$ grey levels); Right: Overdetermined system (k=18 LR observations of 30×30 pixels each), higher noise condition ($\sigma = 1.2$ grey levels). We see that a larger T is preferred in the latter situation, whereas a range of T works well for the former. We also find that the MSE produced with λ estimated adaptively is comparable to that obtained with more informed λ settings. The error bars represent one standard error over 10 independent realisations of the system matrix $\boldsymbol{W}$ and the additive noise.

between the recovered and the true image against varying T. We see that, there is little variation in this case, most likely because the level of noise is small. On this plot, we also superimposed the MSE figures that we could get if λ was set knowing the ground truth image. We see, the obtained MSE values are not very far, in fact comparable to the idealised optimal ones, and the variation with T is also negligible.

The right hand plot repeats the experiment in an overdetermined system and in higher noise conditions. We see, the value of T matters in this case, and there is clearly an optimal range. Three sets of results are superimposed here: with

λ estimated adaptively from the data; with λ estimated under assumption that the noise variance is known, and with λ estimated from the ground truth image. We can see the MSE performances are comparable in this setting as well.

6 Conclusions

We developed a new method for estimating the regularisation parameter for MAP-based resolution enhancement with Huber-MRF priors. This was achieved by upper-bounding the partition function in the Huber density in conjunction with a pseudo-likelihood approach. We demonstrated the added flexibility achieved over the traditional Huber-MRF, and the good performance of our approach in severely under-determined image super-resolution, as well as in overdetermined systems. In future work it will be of interest to compare the adaptive prior presented here with non-convex prior formulations such as [6]. Non-convex priors generally have a better ability to recover texture than convex priors can, but are more difficult to optimise. We should note, despite each iteration of our algorithm solves a convex problem, the marginal prior in eq. (10) is akin a scale-mixture, and is non-convex. Therefore the approach presented here may be seen as breaking down a non-convex prior based optimisation problem into a series of convex ones.

References

1. Geman, S., Gragne, C.: Markov random field image models and their applications to computer vision. In: Proc. Intl Congress of Mathematicians, pp. 1496–1517 (1986)
2. Zibetti, M.V.W., Bazan, F.S.V., Mayer, J.: Determining the regularization parameters for super-resolution problems. Signal Processing 88, 2890–2901 (2008)
3. Hardie, R.C., Barnard, K.J.: Joint MAP Registration and High-Resolution Image Estimation Using a Sequence of Undersampled Images. IEEE Trans. Image Processing 6(12), 621–633 (1997)
4. He, H., Kondi, L.P.: MAP Based Resolution Enhancement of Video Sequences Using a Huber-Markov Random Field Image Prior Model. ICIP (2), 933–936 (2003)
5. He, H., Kondi, L.: Choice of Threshold of the Huber-Markov Prior in Map-based Video Resolution Enhancement. In: Proc. IEEE Canadian Conf. on Electrical and Computer Engineering, Niagara Falls, Canada, vol. II, pp. 801–804 (May 2004)
6. Kabán, A., Ali-Pitchay, S.: Single-frame Image Recovery using a Pearson type VII MRF. Neurocomputing 80, 111–119 (2012)
7. Pickup, L.C., Capel, D.P., Roberts, S.J., Zissermann, A.: Bayesian Methods for Image Super-Resolution. The Computer Journal 52, 101–113 (2009)
8. Buntine, W.L., Weigend, A.S.: Bayesian back-propagation. Complex Systems 5, 603–643 (1991)

Sparse Prototype Representation by Core Sets

Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer

CITEC Centre of Excellence, Bielefeld University, 33615 Bielefeld, Germany
fschleif@techfak.uni-bielefeld.de

Abstract. Due to the increasing amount of large data sets, efficient learning algorithms are necessary. Also the interpretation of the final model is desirable to draw efficient conclusions from the model results. Prototype based learning algorithms have been extended recently to proximity learners to analyze data given in non-standard data formats. The supervised methods of this type are of special interest but suffer from a large number of optimization parameters to model the prototypes. In this contribution we derive an efficient core set based preprocessing to restrict the number of model parameters to $O(\frac{n}{\epsilon^2})$ with n as the number of prototypes. Accordingly, the number of model parameters gets independent of the size of the data sets but scales with the requested precision ϵ of the core sets. Experimental results show that our approach does not significantly degrade the performance while significantly reducing the memory complexity.

1 Introduction

Modern measurement technologies e.g. in the life sciences provide large data sets with multiple sources of domain knowledge, which can be used to optimize the analysis of the data. Accordingly methods which can integrate domain knowledge and are interpretable are of special interest [1]. Prototype algorithms are interpretable models and have been recently extended to simplify domain knowledge integration in different ways. They represent their decisions in terms of typical representatives contained in the input space or by approximations thereof. Prototypes can directly be inspected by human experts in the same way as data points: for example, physicians can inspect prototypical medical cases, prototypical images can directly be displayed on the computer screen, prototypical action sequences of robots can be performed in a robotic simulation, etc. Since the decision in prototype-based techniques usually depends on the similarity of a given input to the prototypes stored in the model, a direct inspection of the taken decision in terms of the responsible prototype becomes possible.

Different algorithms have been proposed to obtain supervised and unsupervised prototype models. One key element is the representation of the data, either by an appropriate metric [2] or by using similarities or dissimilarities as obtained from domain specific measures of proximities. In the latter case the data are represented by an $N \times N$ matrices $\mathcal{P}$ of proximity measures, with N as the number of samples and the prototypes are modeled using a $n \times N$ coefficient matrix, with n as the number of prototypes.

If $\mathcal{P}$ consists of metric similarities, later denoted as $\mathcal{S}$ [1] , we have a kernel matrix and can use standard kernel classifiers like the Support Vector Machine (SVM) [3] or the

[1] If $\mathcal{P}$ are dissimilarities, as e.g. obtained from Euclidean distances we denote the matrix by $\mathcal{D}$.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 302–309, 2013.

Probabilistic Classifier Vector Machine (PCVM) [4]. But, the SVM type algorithms determine models which are not interpretable because the model parameters are extreme cases of the data and hence less interpretable by the domain expert [5]. The PCVM is better to interpret but has an initial runtime complexity of $O(N^3)$ which scales down during learning to $O(m^3)$ with m as the number of effectively used basis functions, using sparsity constraints [4]. The memory complexity of PCVM scales in a similar way from $O(N^2)$ to $O(m^2)$ during the optimization. For proximity data supervised prototype based learning algorithms have in general a runtime complexity of $O(n^2 \cdot N)$ using the recent approximation strategies [6] and a memory complexity of $O(n \cdot N)$, with $n \ll N$. In this paper we will derive a strategy to reduce this memory complexity to $O(\frac{n}{\epsilon^2})$, independent of the number of samples, which obviously also improves the practical runtime for larger data sets, although the theoretical runtime complexity lasts with $O(N)$. We will achieve this by using the concept of core sets [7], already used successfully in supervised learning in [8]. First we will give a short introduction to prototype based learning for proximity data and highlight current problems in this field. Subsequently, we give a brief review of core sets and link both approaches. In the experiments we show that the proposed method is efficient compared with the standard strategies.

2 Prototype Based Learning

Prototype based learning is a very generic approach (see e.g. [9]) with good generalization properties [10]. The prototypes, as the main model parameters are derived as representants of the quantized data space. Accordingly the models are sparse. More recent extension of prototype based approaches for proximity data, represent the prototypes by means of linear combinations of the original data [11], where the models become often very dense [5].

Assume data $\mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$, are given in a D dimensional space. Prototypes are elements $\mathbf{w}_j \in \mathbb{R}^D, j = 1, \ldots, K$, of the same space with $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$, the set of all prototypes of the model. They decompose data into receptive fields $R(\mathbf{w}_j) = \{\mathbf{x}_i : \forall k \; d(\mathbf{x}_i, \mathbf{w}_j) \leq d(\mathbf{x}_i, \mathbf{w}_k)\}$ based on some metric distance measures, like the squared Euclidean distance $d(\mathbf{x}_i, \mathbf{w}_j) = \|\mathbf{x}_i - \mathbf{w}_j\|^2$. The goal of prototype-based machine learning techniques is to find prototypes which represent a given data set as accurately as possible. We will also assume that the data $\mathbf{x}_i$ are equipped with prior class labels $c(\mathbf{x}_i) \in \{1, \ldots, L\}$ in a finite set of priorly known classes in the crisp case or can be associated to a vector of class assignments $l(\mathbf{x}_i) \in [0, 1]^L$, normalized such that $\sum_k^L l(\mathbf{x}_i)^k = 1$ which we will call soft or fuzzy labels. One of the most popular approaches for supervised prototype learning is the Robust Soft LVQ [12], a large margin classifiers [10], optimizing a cost function using gradient descend.

For proximity data an extended RSLVQ was proposed in [13]. One assumes $w_j = \sum_l \alpha_{jl} x_l$ where $\alpha_{jl} > 0$ with $\sum_l \alpha_{jl} = 1$. Then, we can compute for a given data point x_i: $\|x_i - w_j\|^2_{pq} = s_{ii} - 2\sum_l \alpha_{jl} s_{il} + \sum_{ll'} \alpha_{jl}\alpha_{jl'} s_{ll'}$ Hence we can compute distances of all data points and prototypes based on pairwise data *similarities*: $s_{ij} \in \mathcal{S}$ only, in quadratic time. Further, we do not need to represent prototypes w_j explicitly, rather, the coefficients α_{jl} are sufficient. Similarly, we find $\|x_i - w_j\|^2_{pq} = \sum_l \alpha_{jl} d_{il} - 1/2 \cdot \sum_{ll'} \alpha_{jl}\alpha_{jl'} d_{ll'}$ provided $\sum_l \alpha_{jl} = 1$. Hence, as an alternative, we can compute distances via all pairwise *dissimilarities*: $d_{ij} \in \mathcal{D}$ of data in quadratic time. In the following

we will focus on similarities only, but extensions to metric dissimilarities are straight forward using techniques of [14].

The coefficient matrix Γ, with α_{jl} being the l-th coefficient of the j prototype, has most often $n \times N$ entries and is densely populated in the final model. Strategies to improve this by explicit sparsity concepts have been proposed in [5,13] but introduced additional parameters and are hard to control. Also these techniques are often not applicable in the first steps of the optimization such that, similar like for PCVM the initial complexity is high and a large number of parameters needs to be estimated. There are two issues, one is the potentially large squared proximity matrix $\mathcal{P}$ and the second is the dependent matrix Γ. In [15] a concept, known as core sets was used to approach the first problem, we will use similar ideas to approach the second problem, an issue for prototype based models, due to the explicit modeling of the prototypes by data points.

3 Core Sets

The notion of core-sets appears in solving the approximate minimum enclosing ball (MEB) problem in computational geometry [7].

Definition 1 (Minimum enclosing ball problem). *Let* $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \in \mathbb{R}^D$ *a set of points. The objective is to find a minimum enclosing ball with radius* R *and center point* $\mathbf{c}$ *such that* $\|\mathbf{c} - \mathbf{v}_i\|^2 \leq R^2 \; \forall i$. *Hence* $B(c, R) = \{\mathbf{v} | R \geq \|c - \mathbf{v}\|, \mathbf{v} \in V\}$.

As shown in [15], the MEB problem can be equivalently express as a quadratic dual optimization problem: $MEB = \min_{\alpha_i \geq 0, \sum \alpha_i = 1} \alpha K \alpha^\top - \sum_i \alpha_i K(i,i)$, with K the kernel matrix defined on V and $\mathbf{c}$ and $\mathbf{v}$ represented in a kernel space as shown before. The radius R is obtained by solving $R = \sum_i \alpha_i K(i,i) - \alpha K \alpha^\top$. The center is used only in the distance calculation which can be expressed using the kernel trick and is a linear combination of the training points based on the obtained α-vector, $\mathbf{c} = \sum_{i=1} \alpha_i \phi(\mathbf{v}_i)$.

Definition 2 (Core set). *Let* $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \subseteq V$, *then* $Q \subset S$ *is a core set, if* $S \subset B(c, (1+\epsilon)R)$ *and* $B(c, R) = MEB(Q)$.

An encouraging property of core-sets is that the number of elements in it is independent of the data dimensionality and size [7]. Obviously the notion of a MEB ball is closely related to that of a prototype. The prototype, represents a set of points as a receptive field, which borders are constrained by the other prototypes, whereas the MEB represents a set of points, restricted by the radius R and the center position c. The center of the MEB and the prototype position, defined on the same set of points, may not coincide. Nevertheless, a MEB around a set of points in a receptive field of a prototype is often a good approximation for this field.

4 Prototype Models by Core Sets

The idea is to represent a prototype w_j in the matrix Γ by those coefficients α_{jl} which are used to model the MEB of this set of points. This approximation is reasonable as long as the receptive field is close to a sphere, in the similarity space. This is also an

implicit assumption of prominent prototype learning algorithms like the original Robust Soft LVQ [12], where the prototypes are actually modeled as Gaussians with equal bandwidth. We will derive our approach as a pre-processing step prior to any prototype model. In general prototype learning algorithms are initialized, class wise by unsupervised clustering approaches [12], using e.g. k-means or soft competitive learning (SCL) [16] with the number of prototypes per class specified by the user as a meta parameter. For large scale problems novel approximations of k-means or SCL are available, like approximate and core k-means [17,18] or fast SCL [19]. We will focus on the latter one, which is a batch methods using quasi-newton optimization and is efficient also for $N \geq 1e6$, assuming a small number of clusters n. In each case we assume that the training data have been already partitioned by an unsupervised clustering approach into disjunct subsets. Our approach is summarized in Alg. 1

Algorithm 1. Prototype representation by core sets.

1: **init:**
2: Let T be a set of labeled training points and Γ an empty matrix $n \times N$
3: $\epsilon = 0.1$ ▷ Fixed approximation error of the MEB
4: Let $Z = \{z_1, \ldots, z_n\}$ a set of classwise clustered training data
5: $z_j = \{(x_1, l_1)^j, \ldots, (x_k, l_k)^j\}$ using e.g. kmeans
6: **for all** $z_j \in Z$ **do**
7: $[\alpha_j, S_j] := MEB(z_j, \epsilon)$
8: ▷ α-values of the MEB solution for cluster j, S_j core set indices w.r.t. T
9: $\Gamma_j := \alpha_j$
10: ▷ Represent prototype j by the MEB solution for cluster j at indices S_j
11: **end for**
12: Γ := Simplify(Γ) ▷ Reduce Γ to non-empty columns
13: **return** Γ;

The MEB algorithm used in line 7 of Alg. 1 is the same as used in [8]. Starting with two random initial points the MEB is increased in each iteration by the point which is farthest away until the set is covered by the MEB within a circle of $(1+\epsilon) \times R$. A cluster is described by the α values as obtained from the MEB. In general the MEB needs only few iterations such that the number of core set points for a cluster is often in the range of $1 - 10$, theoretically it is bounded by $O(\frac{1}{\epsilon^2})$.

To identify the point which is farthest away in the MEB algorithm, probabilistic sampling is employed [8]. Assume we have a ranking (rank) of the distances of all points to the current center c. We now choose the set F as the 5% points which are furthest away from c, or have a high rank. The probabilistic sampling strategy ensures that a selection of only 59 random points from the training data will contain a point from F with a probability of 95%. The core set S is typically very small $|S| \ll N$ such that the runtime complexity of the calculation of the quadratic optimization problem in the MEB's in Algorithm 1 is given by $O(|S_j|^2)$. For the addressed scale of N we will consider this complexity as sufficient. For very large N it would also be possible to avoid the calculation of a MEB if we know the expected radius r for our problem.

This could be done following concepts of [20] using an enclosing ball concept, but we will not discuss this in the following.

Obviously, the Γ matrix needs not to be defined in advance and may grow during the steps of Algorithm 1 if we add $|S_j| = q_j$, columns to Γ for each prototype because all z_j are disjunct. The obtained matrix Γ can subsequently be used in any prototype based classifier for proximity data as an initial model. Using e.g. a supervised prototype classifier the prototypes can be further optimized to improve the class discrimination. For a prototype j this can be done by modifying the coefficients in the j row of the matrix Γ. As will be exemplary shown in the following this can be done by adapting not only the already used coefficients $\Gamma_{j\cdot}$ but also those not yet used to represent of the prototype due to the disjunct initialization. As an example we consider a checker board problem with 4×4 clusters in $2 - D$. We represent these data by an Euclidean inner product kernel and define two multi-modal classes as shown in Figure 1, such that one class has 5 clusters and the other 11 clusters. The number of prototypes is defined as 7 per class, which is (theoretically) sufficient to discriminate between the classes, using some boundary effects of the data. Since the Γ matrix is initialized class wise and the number of prototypes is 7 per class, the larger class is underrepresented.

The initial post-labeling accuracy is 90%, using the prototype learner presented in [19] we adapt the positions of the prototypes and also the labeling of the prototypes. After 100 iterations we obtain an optimized prototype representation as shown in Figure 1. We observe that the smaller, over represented class, has lost prototypes to the under represented class. These prototypes now show new non-vanishing coefficients in Γ for the indices of the new class they have been assigned to and have also changed their class labeling. The final accuracy is 99% showing that the core-set representation coupled with supervised post training has sufficient representation power.

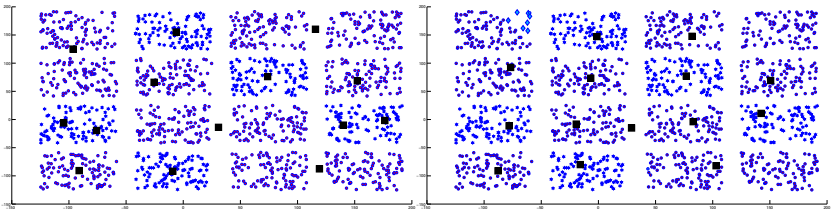


Fig. 1. Checker board data: initial model (left) and after optimization (right) using core sets and a kernel RSLVQ formulation. The top plot shows the data with the given labeling, the right plot the predicted labels on the final model. Initially both classes are modeled by 7 prototypes. The class labels are denoted by different shapes and colors. The prototypes are black squares. Using the initial model a misclassification of 10% is obtained (not illustrated). In the right figure one obtains an accuracy of 99%, few misclassifications can be observed shown as cyan $\diamondsuit$.

5 Experiments

We now show the efficiency of the approach at different data sets, represented by metric proximities, given as a kernel. The number of prototypes was defined such that a reasonable modeling of the data can be achieved guided by the number of classes and prior

Table 1. Experiments with different kernels using the core approach and random selection. Prediction accuracy (Acc), runtime (RT) and memory complexity (used points in Γ (Pts))

Data set	Core			$k\%$			10%		
	Acc	RT	Pts	Acc	RT	Pts	Acc	RT	Pts
Checker	$99.06 \pm 0.52\%$	54.68s	29.4 ± 1.1	66.21 ± 0.94	9.36	31	$99.42 \pm 0.7\%$	74.3s	7200
MNIST	$86.81 \pm 0.5\%$	241s	731.8 ± 3.3	86.13 ± 0.09	113.45	732	$90.37 \pm 1.0\%$	513s	5600
USPS	$87.68 \pm 1\%$	101.6s	714 ± 5.1	$87.51 \pm 1.3\%$	80.9s	714	$88.18 \pm 0.6\%$	95.22s	880
SPAM	$87.48 \pm 2\%$	15.23s	31 ± 1.2	$84.66 \pm 4.1\%$	6.91s	31	$89.72 \pm 0.4\%$	10.71s	369
Phoneme	$83.20 \pm 3.7\%$	14.62s	83.4 ± 0	$75.52 \pm 8.4\%$	9.02s	84	$84.57 \pm 4.0\%$	14.26s	293
Image	$78.62 \pm 1.7\%$	8.2s	27 ± 1.7	$77.7 \pm 1.6\%$	$4.4s$	27	$80.73 \pm 3.9\%$	4.79s	167

experiments of the authors [19]. Please note, that the measure of prediction accuracy on the test data is only used to demonstrate the stability of our approach, we did not do any meta-parameter tuning, although the obtain results are already competitive.

The MNIST data [2] contain $70000, 784$-dimensional binary images from 10 digit classes which have been processed by a neural kernel $k(\mathbf{v_i}, \mathbf{v_j}) = \tanh(av_i^\top v_j + b)$ with $a = 0.0045$ and $b = 0.11$ acc. to [17], we used 10 prototypes per class.

The Checkerboard data is an artificial data set with 90000 $2-$dimensional samples organized on a 3×3 grid with alternating labels in two classes, 5 prototypes have been used per class with a linear kernel to provide the similarities. By construction this data set is highly multimodal.

For all other data sets we use an elm kernel [21], which is a defacto parameter free RBF kernel. USPS [3] contain $11000, 256$-dimensional character feature vectors from 10 classes analyzed by 10 prototypes per class. The SPAM database [4], contain $4601, 57$-dimensional feature vectors in 2 classes which have been modeled by 2 prototypes per class. Further we used the phoneme data set with 3656 points in 20 dimensions with 13, slightly imbalanced classes. For this data set 1 prototype per class was taken. As another example we used the Image data set with 2086 sample in 18 dimensions and 2 classes, modeled by 2 prototypes per class and taken from [22].

All initial clusterings were obtained by fast SCL [19] with a runtime complexity of $O(n^3 \cdot N)$. We evaluate the runtime and memory complexity using our presented core-set approach (core), an alternative representation by selecting a random subset of samples for the prototype modeling with respect to 10% of the data and with $k\%$ where, k is set to get a similar complexity as within the core-set model. As the final prototype classifier we use a full probabilistic formulation of kernel RSLVQ [13,23]. All experiments are done within a $5-$ fold crossvalidation.

The results are shown in Table 1 for multiple data sets, with different data characteristics (number of classes, number of samples, data dimensionality). We also experimented with a different number of columns of the Γ matrix, or different random subsets of representation points, to obtain a similar accuracy than by the core-set approach. The minimal subset size for the Checker data is with around 5% leading to an accuracy of 94% with 301 representation points. For the Phoneme data 5% representation points are

[2] `http://yann.lecun.com/exdb/mnist/`
[3] `http://www.cs.nyu.edu/~roweis/data.html`
[4] `http://archive.ics.uci.edu/ml/datasets`

necessary to get 83% with 147 points. For the MNIST, USPS, SPAM and the Image data a random subset at $k\%$ is already sufficient.

As a general result we find that the proposed approach (*core* column of Table 1) shows similar good prediction results compared to a large model[5], shown in the 10% column. As expected the number of representation points, defined by the core set approach is in general very small, with one exception, for the USPS data where the core-set model is only 20% smaller than the largest considered set.

If we compare our results with a randomly chosen representation set of the same size as the core-set model, shown in column $k\%$, we found that in all cases the core-set model has an equivalent, or in parts substantial better prediction accuracy. For some data sets and using a random selection strategy, the number of representation points has to be substantially increased by a factor of 10 to achieve a similar accuracy.

Depending on the data set a random representation set may substantially degenerate the accuracy of the final model, as shown for the Checker and the Phoneme data set. This can be potentially be explained by the multimodality of these data sets. With respect to the experiments the proposed approach is a systematic and effective strategy to sample a representation set from the training data. For larger data sets a naive random sampling, constrained by the available memory can fail and the obtained models are unnecessary complicated and hard to interpret. Without the information, available here by the core-set approach, it would not easy to guess a reasonable value for $k\%$ but we would rely on adhoc decision like 1% or 10% of the data upper-bounded by the available memory. Using the core-set approach the necessary memory to achieve a very good model representation is $5-10$ times smaller for the considered data than using a adhoc decision for the considered data. Theoretically the memory complexity scales with $O(\frac{n}{\epsilon^2})$, independent of N, such that for very large data sets our strategy is even more interesting.

6 Conclusions

Prototype based models for proximity data are complex and less interpretable for large data sets. This is due to the large coefficient matrix Γ, representing the prototypes by linear combination of original training points. The presented technique automatically identifies a small set of representation points for the modeling of proximity prototypes. The approach achieves competitive prediction accuracy with respect to more complex prototype models. The number of representation points is automatically selected and is independent of the number of training points. The obtained model is not only sparse, with low memory requirements, but also accelerates the model calculation. This effect is most pronounced for large data. The technique can be effectively used to obtain sparse prototype models also for very large proximity data.

Acknowledgements. This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication. Additional funding was provided by the Cluster of Excellence 277 CITEC funded in the framework of the German Excellence Initiative.

[5] A full model, using N points is often too costly, but some experiments with larger representation sets did not significantly improve the accuracy

References

[1] Belle, V.V., Lisboa, P.J.G.: Research directions in interpretable machine learning models. In: Proc. of ESANN 2013 (2013)

[2] Biehl, M., Hammer, B., Schneider, P., Villmann, T.: Metric learning for prototype-based classification. In: Bianchini, M., Maggini, M., Scarselli, F., Jain, L.C. (eds.) Innovations in Neural Information Paradigms and Applications. SCI, vol. 247, pp. 183–199. Springer, Heidelberg (2009)

[3] Vapnik, V.: The nature of statistical learning theory. Statistics for engineering and information science. Springer (2000)

[4] Chen, H., Tino, P., Yao, X.: Probabilistic classification vector machines. IEEE Transactions on Neural Networks 20(6), 901–914 (2009)

[5] Schleif, F.M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype-based classification. Journal of Neural Systems 21(6), 443–457 (2011)

[6] Gisbrecht, A., Mokbel, B., Schleif, F.M., Zhu, X., Hammer, B.: Linear time relational prototype based learning. Journal of Neural Systems (2012) (in press)

[7] Badoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: STOC, pp. 250–257 (2002)

[8] Tsang, I.H., Kocsor, A., Kwok, J.Y.: Large-scale maximum margin discriminant analysis using core vector machines. IEEE TNN 19(4), 610–624 (2008)

[9] Schneider, P., Biehl, M., Hammer, B.: Distance learning in discriminative vector quantization. Neural Computation 21(10), 2942–2969 (2009)

[10] Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of lvq algorithms. Journal of Machine Learning Research 8, 323–360 (2007)

[11] Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation 22(9), 2229–2284 (2010)

[12] Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Computation 15(7), 1589–1604 (2003)

[13] Hammer, B., Hoffmann, D., Schleif, F.M.: Learning vector quantization for (dis-)similarities. NeuroComputing (in press, 2013)

[14] Schleif, F.-M., Gisbrecht, A.: Data analysis of (Non-)Metric proximities at linear costs. In: Hancock, E., Pelillo, M. (eds.) SIMBAD 2013. LNCS, vol. 7953, pp. 59–74. Springer, Heidelberg (2013)

[15] 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)

[16] Martinetz, T., Schulten, K.: Topology representing networks. Neural Networks 7(3), 507–522 (1994)

[17] Chitta, R., Jin, R., Havens, T., Jain, A.: Approximate kernel k-means: Solution to large scale kernel clustering, pp. 895–903 (2011)

[18] Har-Peled, S., Kushal, A.: Smaller coresets for k-median and k-means clustering. Discrete & Computational Geometry 37(1), 3–19 (2007)

[19] Schleif, F.M., Zhu, X., Gisbrecht, A., Hammer, B.: Fast approximated relational and kernel clustering. In: Proceedings of ICPR 2012, pp. 1229–1232. IEEE (2012)

[20] Tsang, I.W., Kocsor, A., Kwok, J.T.: Simpler core vector machines with enclosing balls. In: Proc. of ICML 2007, pp. 911–918 (2007)

[21] Frénay, B., Verleysen, M.: Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing 74(16), 2526–2531 (2011)

[22] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A.J., Müller, K.R.: Invariant feature extraction and classification in kernel spaces. In: Solla, S.A., Leen, T.K., Müller, K.R. (eds.) NIPS, pp. 526–532. The MIT Press (1999)

[23] Schneider, P., Geweniger, T., Schleif, F.M., Biehl, M., Villmann, T.: Multivariate class labeling in robust soft LVQ. In: Proceedings of ESANN 2011, pp. 17–22 (2011)

Reconstruction of Wind Speed Based on Synoptic Pressure Values and Support Vector Regression

B. Saavedra-Moreno[1], S. Salcedo-Sanz[1], L. Carro-Calvo[1], A. Portilla-Figueras[1], and J. Magdalena-Saiz[2]

[1] Universidad de Alcalá, Alcalá de Henares, Madrid, Spain
[2] Solute Ingenieros, Spain

Abstract. Wind speed reconstruction is a challenging problem in areas (mainly wind farms) where there are very few direct wind measurements available. In this paper we discuss a methodology for wind speed reconstruction in wind farms from pressure measurements. Specifically, we tackle the problem of wind speed prediction from synoptic pressure patterns by considering a regression problem between a grid of sea-level pressure values and a wind speed module measure. The performance of a Support Vector Regression algorithm is analyze, together with the inclusion of a genetic algorithm in order to perform a feature selection for the input variables (pressure). Results considering real wind speed data in Boston, USA, with sea-level pressure values from two alternative global meteorological models (NOAA and ECMWF) are presented.

1 Introduction

Wind speed reconstruction and prediction are two of the most most important problems related to wind energy. These problems are associated with different important decisions to take in wind farms, such as production analysis, planning or even micro-sitting of new wind turbines, etc. Existing approaches for these problems are mainly based on historic data and wind measurements, from which statistical models are constructed in order to explain the wind behavior [1]. The problem with this approach based on wind measurements is that, in some cases, these data are not available, due to fails in the measurement systems, or just because the terrain is a prospective site to install a wind farm, and there is not a meteorological tower installed yet. In these problematic cases, the possibility of obtaining wind speed reconstructions from indirect measurements such as sea-level pressure is currently a hot topic, in which many renewable energy companies are investing lots of resources. In this sense, different recent works have used synoptic pressure as indirect measurements to study different atmospheric phenomenons including wind speed reconstruction [2].

In this paper, the problem of wind speed reconstruction in a given point (wind farm), from a related synoptic sea-level pressure values in a grid is tackled. The problem involves daily pressure patterns in a synoptic grid and a wind speed

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 310–317, 2013.

module measure. We consider a regression problem in which the exact value of the wind speed must be estimated (reconstructed) from pressure values. Specifically, we describe the application of a Support Vector Regression algorithm to the problem, in which the sea-level values of synoptic-pressure are the input (predictive) variables. We also analyze the effect of a feature selection in the pressure space by means of a genetic algorithm. The performance of the proposed approach is evaluated in a real wind speed reconstruction problem with data from a wind farm in the East coast of the USA.

The structure of the rest of the paper is the following: next section presents the definition of the problem and also reviews the main characteristics of the SVMr considered in this problem. The experimental part of the paper is then presented in Section 3. Finally, Section 4 closes the paper giving some concluding remarks.

2 Problem Formulation and Regressor Considered

The problem considered in this paper can be summarized as follows: Let $\mathbf{d}_t,\ t = 1\ldots,T$, be a series of wind speed (module), measured in a given point (a wind farm in this case), for a given period of time T. Let $P_t,\ t = 1\ldots,T$, be a series of synoptic-scale pressure measurements in a given grid. In our case, each component of P_t is a matrix of $M \times N$ surface pressure values, measured in a grid surrounding the wind farm location. Both, the time interval of reconstruction and the range (dimensions and resolution) of the pressure grid are parameters to be fix in the problem. The problem consists of obtaining the relationship between the wind speed and the synoptic-scale pressure situation using a given regressor, in such a way that a measure of error between the wind speed reconstruction and a real value is minimized. Specifically, we consider the Support Vector Regression algorithm as the regressor to be used in the problem.

2.1 ϵ-SVM Formulation

One of the most important forecasting statistic models are the support vector machines for regression (SVMr) [3]. The SVMr are used in a large amount of regression problems, including wind speed prediction [4]. The SVMr uses kernel theory to increase the quality of regression models and, in most cases can be solved as a convex optimization problem. The SVMr does not only take into consideration the prediction error of the data, but also the generalization of the model, i. e., its capability to improve the prediction of the model when new data are evaluated by it (avoiding over-fitting). Also there are several versions of SVMr, in this case the classical model presented in [3] is described.

The ϵ-SVM method for regression [3] consists of, given a set of training vectors $\mathcal{S} = \{(\mathbf{x_i}, y_i), i = 1,\ldots,l\}$, training a model of the form $y(\mathbf{x}) = f(\mathbf{x}) + b = \mathbf{w}^T\phi(\mathbf{x}) + b$, to minimize a general risk function of the form

$$R[f] = \frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{1}{2}C\sum_{i=1}^{l} L\left(y_i, f(\mathbf{x})\right) \tag{1}$$

where C is the SVM regularization parameter (that must be set in advance), $\mathbf{w}$ controls the smoothness of the model, $\phi(\mathbf{x})$ is a function of projection of the input space to the feature space, b is a parameter of bias, $\mathbf{x_i}$ is a feature vector of the input space with dimension N, y_i is the output value to be estimated and $L\left(y_i, f(\mathbf{x})\right)$ is the loss function selected. In this paper, we use the L1-SVR (L1 support vector regression), characterized by an ϵ-insensitive loss function [5]

$$L\left(y_i, f(\mathbf{x})\right) = |y_i - f(\mathbf{x_i})|_\epsilon \tag{2}$$

In order to train this model, it is necessary to solve the following optimization problem [5]:

$$\min\left(\frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \tag{3}$$

subject to

$$y_i - \mathbf{w}^T\phi(\mathbf{x_i}) - b \leq \epsilon + \xi_i, \quad i = 1, \ldots, l \tag{4}$$

$$-y_i + \mathbf{w}^T\phi(\mathbf{x_i}) + b \leq \epsilon + \xi_i^*, \quad i = 1, \ldots, l \tag{5}$$

$$\xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, l \tag{6}$$

The dual form of this optimization problem is usually obtained through the minimization of the Lagrange function, constructed from the objective function and the problem constraints. In this case, the dual form of the optimization problem is the following:

$$\max\left(-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x_i}, \mathbf{x_j}) - \right.$$
$$\left. -\epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*)\right) \tag{7}$$

subject to

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{8}$$

$$\alpha_i, \alpha_i^* \in [0, C] \tag{9}$$

In addition to these constraints, the Karush-Kuhn-Tucker conditions must be fulfilled, and also the bias variable, b, must be obtained. We do not detail this process for simplicity, the interested reader can consult [5] for reference. In the dual formulation of the problem the function $K(\mathbf{x_i}, \mathbf{x_j})$ is the kernel matrix, which is formed by the evaluation of a kernel function, equivalent to the dot

product $\langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$. A usual election for this kernel function is a Gaussian function, as follows:

$$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma \cdot \|\mathbf{x_i} - \mathbf{x_j}\|^2}. \tag{10}$$

The final form of function $f(\mathbf{x})$ depends on the Lagrange multipliers α_i, α_i^*, as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(\mathbf{x_i}, \mathbf{x}) \tag{11}$$

The well-known LibSVM implementation of the algorithm in C [6] has been considered in this paper.

3 Experiments and Results

This section presents the experimental part of this study. In order to show the performance of the proposed approach, several experiments have been carried out. First, the available data and the methodology of the experiments are described. Then, the results obtained by applying the SVMr and a genetic algorithm for feature selection are shown.

3.1 Available Data for the Study and Methodology

The proposed wind speed reconstruction system, with the methodology described above, has been tested by using a public database of wind speed measurements (wind speed at 10m), maintained by the Wind Energy Center of the University of Massachusetts (UMWEC). Specifically, a location with a significant amount of available data has been chosen, in order to split data into a train and a test set. The chosen location is Thompson Island, Boston, with data from 2001 to 2012. The years with the fewest gaps of measured data have been used: 2001, 2002 and 2003 for the training process, and 2005, 2006 and 2007 for the test period. Figure 1 shows the location of the wind measuring tower considered. This location is clearly influenced by the synoptic-pressure situation, so the reconstruction offered by the proposed system is expected to be good. In order to delimit the size of the pressure grid, a 10° square grid is considered (centered in the location, with 5° in each direction in order to adjust to the minimum resolution given by the atmospheric models), as a compromise solution between the error obtained and the amount of data provided as entries of the algorithm (that influences the training time).

Wind speed averages over 6, 12 or 24 hours are considered to obtain data vectors $\mathbf{d}_t$. On the other hand, average over the same period of time pressure maps are obtained from two climatological databases: National Oceanic and Atmospheric Administration (NOAA) and European Centre for Medium-Range Weather Forecast (ECMWF). The NOAA database provides data from the National Center for Environmental Prediction/National Center for Atmospheric

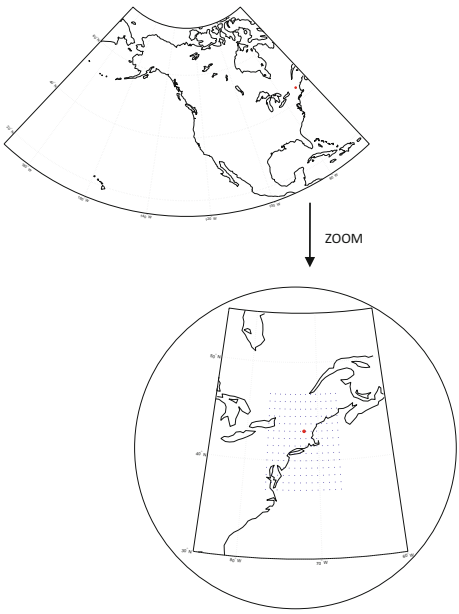


Fig. 1. Wind farm location and grid (0.75°) considered

Research Reanalysis Project (NCEP/NCAR) [7], which are public data profusely used in climatology and meteorology applications. Synoptic-scale sea-level pressure data is available all over the world from 1948/1/1 to the current date, with 4 daily measurements (at 0 am, 6 am, 12 pm and 18 pm) and a latitude and longitude resolution of 2.5°. On the other hand, the ECMWF database provides data from the ERA-Interim reanalysis, which is the latest global atmospheric reanalysis produced by the ECMWF [8]. ERA-Interim covers the period from 1 January 1989, and continues to be extended in near-real time [9], providing data of synoptic-scale pressure with 4 daily measures. Two spacial resolutions are available with this model, 2.5° or 0.75°. We will keep the higher resolution in this case.

Regarding the methodology used in this study, first it is necessary to assign a wind speed value to each synoptic pressure situation. Since wind speed data are measured with a 10-minute interval and pressure data has a 6-hour interval, mean wind speed values have to be calculated, in such a way that we obtain mean wind speed values at 0 am, 6 am, 12 pm, and 18 pm (hours of sea-level pressure measurement). To do so, data from the 3 previous hours, the current hour and the two following hours are used to calculate the mean wind speed value. Then, we have to match the mean wind speed values calculated $\mathbf{d}_t$ with the corresponding synoptic pressure situation P_t. Only instants of time with a valid wind speed value are considered, thus it can exist gaps in the measured wind speed series.

We first consider the complete sea-level pressure data as inputs for the SVMr. We also consider a second case, in which we will try to reduce the number of input data of the SVMr (which can be huge in some cases, for example if spatial resolution of 0.75° is used), by means of the application of a genetic algorithm. In this case the genetic algorithm will search for the best set of reduced points of sea-level pressure, that maintains (or even improve) the performance of the system. Note that by reducing the number of input variables we will obtain a faster training of the SVMr in any case. A classical binary encoded genetic algorithm is considered, in which '0' stands for a non-used variable, and '1' means that the variable has to be used as input in the SVM. The fitness function considered is the prediction error obtained from the SVMr (with fixed parameters), using as input data the variables indicated by the individual, using a cross-validation process with the training data. We have considered classical implementation of crossover and mutation operators in the genetic algorithm, with a tournament selection [10].

In order to have a comparison reference algorithm, we have use the *Multivariate adaptive regression splines* algorithm (MARS), which is a non-parametric regression technique that automatically models non-linearities and interactions between variables, and that has been successfully applied to other real regression problems [11]. We have applied the MARS algorithm by using an open Matlab source freely available in the Internet [12].

3.2 Results

Table 1 shows the results obtained in the problem of wind reconstruction from pressure values considered. This table shows the RMSE obtained in the test set, with the SVMr and the SVMr with feature selection using the GA. Results are given for pressure data from ECMWF and NOAA models. In the case of ECMWF data, the SVMr+GA obtains better result than the SVMr on its own. Both algorithms obtain better results than the MARS approach, with the two considered physical models data. Note that the inclusion of a feature selection process in the system helps improve the SVMr results, for the ECMWF model. On the other hand, this effect is not found in the case of NOAA data, obtaining almost the same result with and without the feature selection with the GA. The explanation to this result is in the grid considered, whereas in the ECMWF a 0.75° is considered, the NOAA data are provided in a less resolution one 2.5°. This means that the number of nodes in the grid is smaller in the NOAA case, so the effect of feature selection is also smaller. On the other hand, in the case of the ECMWF data, the number of nodes in the grid is large, and it seems that the reconstruction system is improved by adjusting (reducing) this number (that is equivalent to reduce the number of pressure values considered as inputs of the SVMr). We have found that the MARS algorithm is computationally faster than the SVMr. The MARS algorithm takes about 20 seconds to complete the training considering the complete grid, whereas the SVMr takes some minutes. However, the quality of the wind speed reconstruction is much better with the SVMr approach.

Note that, following Table 1, the RMSE of the system in reconstruction is low, what means that the reconstruction system is working quite reasonably. The proposed system provides a way of reconstructing wind speed in a given point if we have a wind speed series measured in that point, just by using sea-level pressure values, that are available world-wide thank to the global prediction models existing. Figure 2 shows an example of the wind speed reconstruction in the test (SVMr+GA). Note the excellent reconstruction provided by the system, that is able even to detect strong variations in the 6-h time horizon wind speed reconstruction considered.

Table 1. RMSE (m/s) obtained for the chosen location applying the proposed algorithms, for 6-hours time horizon reconstruction

Data source	SVMr	SVMr+GA	MARS
ECMWF	1.78059	1.76251	2.2694
NOAA	1.83063	1.83543	2.2714

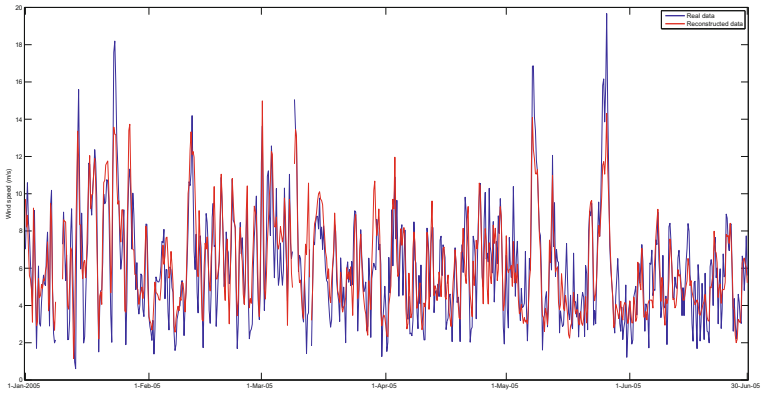


Fig. 2. Wind speed reconstruction with the SVMr+GA approach in part of the considered test set (January to June 2005)

4 Conclusions

In this paper we have considered a new approach for 6-h time horizon wind speed reconstruction based on synoptic pressure measures with a Support Vector regression algorithm. The problem has been stated as a pure regression task in which sea-level synoptic pressure measures act as input (predictive) variables, and an measure of wind speed is used to train the system. A feature selection mechanism with a genetic algorithm has been also considered in the system, with success. Results using wind speed values in Boston, USA, with sea-level pressure values from two different global physic models have shown the good performance of our proposal.

Acknowledgements. This work has been partially supported by Spanish Ministry of Science and Innovation, under project number ECO2010-22065-C03-02. This work is also supported by Solute Ingenieros, through the Project "Desarrollo de nuevos sistemas de caracterización del recurso eólico, evaluacin de emplazamientos específicos y predicción de viento: ETSWIND".

References

1. Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., Feitosa, E.: A review on the young history of the wind power short-term prediction. Renewable and Sustainable Energy Reviews 12, 1725–1744 (2008)
2. Carro-Calvo, L., Salcedo-Sanz, S., Kirchner-Bossi, N., Portilla-Figueras, A., Prieto, L., Garcia-Herrera, R., Hernández-Martín, E.: Extraction of synoptic pressure patterns for long-term wind speed estimation in wind farms using evolutionary computing. Energy 36, 1571–1581 (2011)
3. Smola, A.J., Murata, N., Schölkopf, B., Muller, K.: Asymptotically optimal choice of ϵ-loss for support vector machines. In: Proc. of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing (1998)
4. Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, A.A.: Support vector machines for wind speed prediction. Renewable Energy 29(6), 939–947 (2004)
5. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing (1998)
6. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
7. `http://www.esrl.noaa.gov/psd/data/reanalysis/reanalysis.shtml`
8. `http://data-portal.ecmwf.int/data/d/interim_full_daily`
9. Dee, D.P., Uppal, S.M., Simmons, A.J., et al.: The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Quarterly Journal of the Royal Meteorological Society 137(656), 553–597 (2011)
10. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing, 1st edn. Natural Computing Series. Springer (2003)
11. Zhang, W.G., Goh, A.: Multivariate adaptive regression splines for analysis of geotechnical engineering systems. Computers and Geotechnics 48, 82–95 (2013)
12. Jekabsons, G.: ARESLab: Adaptive Regression Splines toolbox for Matlab (2011), `http://www.cs.rtu.lv/jekabsons/`

Direct Solar Radiation Prediction Based on Soft-Computing Algorithms Including Novel Predictive Atmospheric Variables

S. Salcedo-Sanz[1], C. Casanova-Mateo[2], A. Pastor-Sánchez[1], D. Gallo-Marazuela[1], A. Labajo-Salazar[3], and A. Portilla-Figueras[1]

[1] Universidad de Alcalá, Alcalá de Henares, Madrid, Spain
sancho.salcedo@uah.es
[2] Universidad de Valladolid, Valladolid, Spain
[3] Former Head of the radiometric and atmospheric composition networks Spanish Meteorogical State Agency (AEMET)

Abstract. In this paper we tackle a problem of solar radiation prediction with Soft-Computing Techniques. We introduce new atmospheric input variables in the problem, which help to obtain an accurate prediction of solar radiation. We test the performance of two state-of-the art algorithms: Extreme Learning Machines and Support Vector regression algorithms, in a real problem of solar radiation prediction in Murcia, Spain, where we have obtained excellent results with the proposed techniques.

1 Introduction

Direct Solar Radiation (DSR) estimation is an important problem with direct applications in renewable energy. Solar is one of the most important green sources of energy, that is currently under expansion in many countries of the world, specially in those with more solar potential, such as mid-east and southern Europe countries. An accurate estimation of the energy production in solar energy systems involves the accurate prediction of the DSR, depending on different atmospheric variables [1].

Different computational methods have been recently applied to the problem of solar radiation prediction (both direct and global measures), including Soft-Computing approaches [2]. Specifically, neural networks have been massively applied in this problem, like in [3], where it is shown that radial basis function neural networks obtain excellent performance in the estimation of solar radiation. In [4] a comparison between MLPs and RBF neural networks in a problem of solar radiation estimation is carried out. Experiments in eight stations in Oman show the good results obtained with the neural algorithms. A similar approach, also comparing MLPs and RBFs (with different predictive variables) has been recently proposed in [5]. In this case, the authors test the neural network with data obtained in Iran. Other approaches that also use neural networks as prediction methodology, include novel predictive variables, such as satellite

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 318–325, 2013.

data [6] or temperature and relative humidity [7]. Alternative Soft-Computing algorithms, such as Support Vector Machines (SVM) have been also applied to solar radiation prediction problems from meteorological predictive variables [8]. Specifically, a least-square SVM is proposed in that work, comparing the results obtained with that of auto-regressive and RBF neural networks.

The aim of this paper is twofold. First, we tackle a problem of DSR prediction from atmospheric variables using Soft-Computing techniques that have not been tested, to the best of our knowledge, in this problem. Specifically an Extreme Learning Machine and a ϵ-Support Vector regressor algorithm are applied. Second, we explore the effect of including new atmospheric predictive variables as inputs of the algorithms tested. Measured variables such as the total ozone content of the atmosphere or a measure of particles are included together with variables from atmospheric models such as the prediction of cloudiness in the zone under study. In the experimental part of the paper we show how the proposed model and algorithms are able to successfully solve this solar radiation problem.

The structure of the rest of the paper is the following: next section presents the problem formulation and describes the input variables involved on it. Section 2 briefly summarizes the description of the ELM and SVMr considered in this paper. Section 3 presents the main results obtained in a real solar radiation problem, in Murcia, Spain. Finally, Section 4 closes the paper by giving some concluding remarks.

2 Methods

This section presents the Soft-Computing methods applied in the problem of radiation prediction considered. Basically, the Extreme Learning Machine and the Support Vector Regression algorithm are briefly described.

2.1 Extreme Learning Machines

An ELM is a novel and fast learning method based on the structure of multi-layer perceptrons, recently proposed in [9] and applied thereafter to a large number of classification and regression problems. The most significant characteristic of the ELM training is that it is carried out just by randomly setting the network weights, and then obtaining the inverse of the hidden-layer output matrix. The advantages of this technique are its simplicity, which makes the training algorithm extremely fast, and also its outstanding performance when compared to avantgarde learning methods, usually better than other established approaches such as classical multi-layer perceptrons or support vector machines. Moreover, the universal approximation capability of the ELM network, as well as its classification capability, have been already proven [10,11].

The ELM training method can be defined in the following way: Given a training set $\aleph \triangleq \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbb{R}^n, \mathbf{t}_i \in \mathbb{R}, i = 1, \ldots, N_T\}$, an activation function $g(x)$ and a number of hidden nodes ($\tilde{N}$), the ELM algorithm is summarized in a number of steps:

1. Randomly assign inputs weights $\mathbf{w}_i$ and bias b_i, with $i = 1, \ldots, \tilde{N}$.
2. Calculate the $N_T \times \tilde{N}$ hidden-layer output matrix $\mathbf{H}$, defined as

$$\mathbf{H} \triangleq \begin{bmatrix} g(\mathbf{w}_1 \mathbf{x_1} + b_1) \cdots g(\mathbf{w}_{\tilde{N}} \mathbf{x_1} + b_{\tilde{N}}) \\ \vdots \qquad \ddots \qquad \vdots \\ g(\mathbf{w}_1 \mathbf{x_{N_T}} + b_1) \cdot g(\mathbf{w}_{\tilde{N}} \mathbf{x_{N_T}} + b_{\tilde{N}}) \end{bmatrix}. \tag{1}$$

3. Calculate the output weight vector β as

$$\beta = \mathbf{H}^{\dagger} \mathbf{T}, \tag{2}$$

where $\mathbf{H}^{\dagger}$ stands for the Moore-Pennrose inverse of matrix $\mathbf{H}$ [9], and $\mathbf{T} \triangleq [t_1, \ldots, t_{N_T}]^T$ is the training output vector.

Note that the number of hidden nodes $\tilde{N}$ is a free parameter of the ELM training, and must be estimated for obtaining good results. Usually, scanning a range of $\tilde{N}$ values is the most practical solution for this problem. It is well known that the ELM is an algorithm with a low computational complexity because it just involves the calculation of the output weights by means of the Moore-Pennrose matrix and other minor calculations.

2.2 Support Vector Regression

The ϵ-SVM method for regression [12] consists of, given a set of training vectors $C = \{(\mathbf{x_i}, y_i), i = 1, \ldots, l\}$, training a model of the form $y(\mathbf{x}) = f(\mathbf{x}) + b = \mathbf{w}^T \phi(\mathbf{x}) + b$, to minimize a general risk function of the form

$$R[f] = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{2} C \sum_{i=1}^{l} L\left(y_i, f(\mathbf{x})\right) \tag{3}$$

where $\mathbf{w}$ controls the smoothness of the model, $\phi(\mathbf{x})$ is a function of projection of the input space to the feature space, b is a parameter of bias, $\mathbf{x_i}$ is a feature vector of the input space with dimension N, y_i is the output value to be estimated and $L\left(y_i, f(\mathbf{x})\right)$ is the loss function selected. In this paper, we use the L1-SVR (L1 support vector regression), characterized by an ϵ-insensitive loss function [13]

$$L\left(y_i, f(\mathbf{x})\right) = |y_i - f(\mathbf{x_i})|_{\epsilon} \tag{4}$$

In order to train this model, it is necessary to solve the following optimization problem [13]:

$$\min \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \right) \tag{5}$$

subject to

$$y_i - \mathbf{w}^T \phi(\mathbf{x_i}) - b \leq \epsilon + \xi_i, \quad i = 1, \ldots, l \tag{6}$$

$$-y_i + \mathbf{w}^T \phi(\mathbf{x_i}) + b \leq \epsilon + \xi_i^*, \quad i = 1, \ldots, l \tag{7}$$

$$\xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, l \tag{8}$$

The dual form of this optimization problem is usually obtained through the minimization of the Lagrange function, constructed from the objective function and the problem constraints. In this case, the dual form of the optimization problem is the following:

$$\max \left(-\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\mathbf{x_i}, \mathbf{x_j}) - \right.$$
$$\left. -\epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i (\alpha_i - \alpha_i^*) \right) \tag{9}$$

subject to

$$\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \tag{10}$$

$$\alpha_i, \alpha_i^* \in [0, C] \tag{11}$$

In addition to these constraints, the Karush-Kuhn-Tucker conditions must be fulfilled, and also the bias variable, b, must be obtained. We do not detail this process for simplicity, the interested reader can consult [13] for reference. In the dual formulation of the problem the function $K(\mathbf{x_i}, \mathbf{x_j})$ is the kernel matrix, which is formed by the evaluation of a kernel function, equivalent to the dot product $\langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$. A usual election for this kernel function is a Gaussian function, as follows:

$$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma \cdot \|\mathbf{x_i} - \mathbf{x_j}\|^2}. \tag{12}$$

The final form of function $f(\mathbf{x})$ depends on the Lagrange multipliers α_i, α_i^*, as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(\mathbf{x_i}, \mathbf{x}) \tag{13}$$

3 Experimental Part

3.1 Proposed Predictive Model: Meteorological Variables for Solar Radiation Estimation

It has been recently reported that Clearness Index (horizontal global irradiance/horizontal extraterrestrial irradiance) and Relative Air Mass are the more

relevant input variables to the neural network in problems of solar radiation prediction [2,14]. To some degree or another, both parameters are related to the two processes involved in the solar radiation extinction: scattering and absorption. Considering this, and taking into account that the aim of this study was to predict direct solar irradiance, we assume that the regressor techniques considered would work more accurately if they are trained with parameters related to atmospheric scattering and absorption processes. All the parameters used in our study ranges from the 1st January 2010 to 31st December 2011.

Scattering is a physical process by which a particle in the path of an electromagnetic wave continuously abstract energy from the incident wave and re-radiates that energy in all directions. In the atmosphere one of the most important particles responsible for scattering are aerosols. They are known to be produced by natural processes (volcanic dust, particles from sea spray, wind-blown dust, etc.) as well as by human activity [15]. Its concentration varies with locality and it generally decrease rapidly with height in the troposphere. Then, in order to take into account the presence of atmospheric aerosols, we consider the daily mean aerosol optical depth product obtained from a Cimel CE318 sunphotometer as input parameter. This instrument makes direct sun measurements at wavelengths 340, 380, 440, 500, 670, 870 and 1020 nm. with a field of view of 1.2 degrees. The instrument belongs to the National State Agency of Spain (AEMET), is located in the radiometric observatory of Murcia (38.0° N, 1.2° W) and is part of the NASA Aerosol Robotic Network (AERONET) [16].

Ozone concentration can be derived from Brewer spectrophotometer measurements. Thanks to this instrument it is possible to derive total ozone amount from the ratio of measured sunlight intensities at five wavelengths between 306 and 320 nm with a resolution of 0.6 nm, where the absorption by ozone presents large spectral structures [17]. Thus, we have used the daily mean ground-based total ozone amount derived from the Murcia Brewer spectrophotometer.

Water vapour present in the atmosphere could be considered by selecting the total precipitable water (the amount of liquid water, in mm, if all the atmospheric water vapour in the column were condensed) product derived from an atmospheric sounding as an input parameter. Fortunately the radiometric observatory of Murcia also hosts an upper-air sounding station. Although an atmospheric sounding is launched every twelve hours (00:00 and 12:00 UTC) we have calculated the mean TPW value for every two soundings in order to have the same temporal resolution as the Cimel and Brewer data. Murcia total precipitable water data are freely available on Internet [18].

For a predicting operative approach, it is necessary to contemplate the presence of clouds, because its existence over the area of study clearly affects the amount of direct solar irradiance that reaches a ground surface [2]. For that, we have used data from the numerical weather prediction model GFS (Global Forecast System) maintained by the National Center for Environmental Prediction (USA).

Finally, the theoretical extraterrestrial solar irradiance calculated with the classical equations [19] has been also considered as an input variable. And the real direct solar irradiance (target) that reaches the ground was obtained calculating the daily mean value from the measurements of a pyrheliometer mounted on an automatic solar tracker. This instrument is property of AEMET and is also located in the radiometric observatory of Murcia.

3.2 Results

The available data of direct solar radiation (objective) and predictive meteorological variables where divided into a training and test set (80% and 20% of the data, respectively). The ELM and SVM algorithms were trained using the training set. The ELM is not a deterministic algorithm, but it obtains slightly different results depending on the run, because its random weights structure in the input layer. We have therefore launched it 30 times and the average value has been kept. The SVM with grid search for parameters estimation obtain one value for prediction. Table 1 shows the Mean Square Error of the ELM and SVM prediction, in the test set. The value of R^2 and the training time needed for ELM and SVM approaches are also shown in this table. It is easy to see that the SVM obtains a solution slightly more accurate than the ELM. On the other hand, the ELM training time is extremely fast, much better than the SVM training time (that includes parameters search).

Table 1. Results obtained by the ELM and SVM in the problem of direct radiation prediction. Variance (over 30 ELM runs) is shown in the ELM result

Algorithm	MSE (kJ/m^2)	R^2	training time (s)
ELM	10.60 ± 1.24	0.87	0.1
SVM	9.35	0.91	79.02

Figures 1 and 2 show a comparison between the real direct solar radiation measured and the prediction obtained by the ELM and SVM algorithms, respectively. It can be seen how both approaches are able to follow the trend and some the largest peaks of radiation registered.

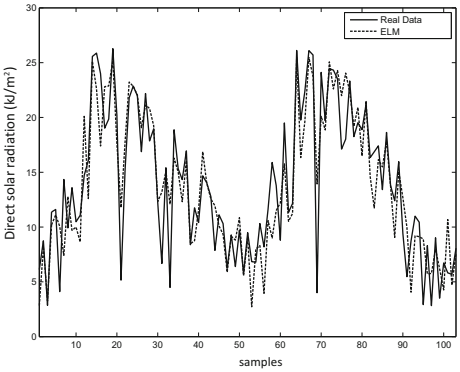


Fig. 1. Real radiation and ELM prediction in the test set considered

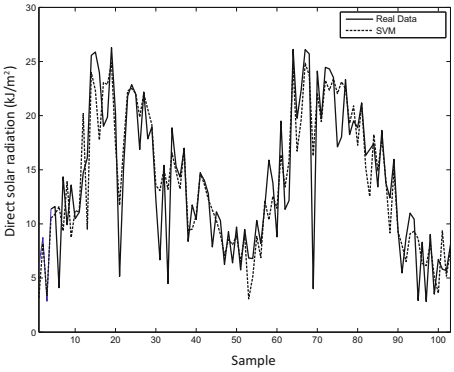


Fig. 2. Real radiation and SVM prediction in the test set considered.

4 Conclusions

In this paper we have tackled a problem of solar radiation prediction with Soft-Computing algorithms. Specifically Extreme Learning Machines and ϵ-Support Vector regression algorithms are tested in this problem. We have proposed a new prediction model where several novel meteorological input variables are considered, such as the total ozone content or the cloudiness prediction in the area of study, from a meteorological model. Experiments in a real problem of solar radiation prediction in Murcia, Spain, have shown the excellent performance of the proposed algorithms. Future lines of research aims at using this prediction system in the frame of smart grids and isolated energy systems, where an accurate prediction of renewable generation is key in order to save costs and improve the performance of systems.

Acknowledgements. This work has been partially supported by Spanish Ministry of Economy, under project number ECO2010-22065-C03-02. The authors would like to thank the Spanish Meteorological State Agency (AEMET) for providing the data used in this study.

References

1. Khatib, T., Mohamed, A., Sopian, K.: A review of solar energy modeling techniques. Renewable and Sustainable Energy Reviews 16, 2864–2869 (2012)
2. Mellit, A., Kalogirou, S.A.: Artificial intelligence techniques for photovoltaic applications: A review. Progress in Energy and Combustion Science 34(5), 574–632 (2008)
3. Benghanem, M., Mellit, A.: Radial Basis Function Network-based prediction of global solar radiation data: Application for sizing of a stand-alone photovoltaic system at Al-Madinah, Saudi Arabia. Energy 35(9), 3751–3762 (2010)
4. Dorvlo, A.S., Jervase, J.A., Al-Lawati, A.: Solar radiation estimation using artificial neural networks. Applied Energy 71(4), 307–319 (2002)
5. Behrang, M.A., Assareh, E., Ghanbarzadeh, A., Noghrehabadi, A.R.: The potential of different artificial neural network (ANN) techniques in daily global solar radiation modeling based on meteorological data. Solar Energy 84(8), 1468–1480 (2010)
6. Senkal, O., Kuleli, T.: Estimation of solar radiation over Turkey using artificial neural network and satellite data. Applied Energy 86(78), 1222–1228 (2009)
7. Rehman, S., Mohandes, M.: Artificial neural network estimation of global solar radiation using air temperature and relative humidity. Energy Policy 36(2), 571–576 (2008)
8. Zeng, J., Qiao, W.: Short-term solar power prediction using a support vector machine. Renewable Energy 52, 118–127 (2013)
9. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70(1), 489–501 (2006)
10. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks 17(4), 879–892 (2006)
11. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multi-class classification. IEEE Transactions on Systems, Man and Cybernetics, Part B 42(2), 513–529 (2012)
12. Smola, A.J., Murata, N., Scholkopf, B., Muller, K.: Asymptotically optimal choice of ϵ-loss for support vector machines. In: Proc. of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing (1998)
13. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing (1998)
14. Lopez, G., Batlles, B., Tovar-Pescador, J.: Selection of input parameters to model direct solar irradiance by using artificial neural networks. Energy 30, 1675–1684 (2005)
15. Liou, K.N.: An introduction to atmospheric radiation. International Geophysics Series (2002)
16. Holben, B.N., Eck, T.F., Slutsker, I., Tanré, D., Buis, J.P., et al.: AERONET-A federated instrument network and data archive for aerosol characterization. Remote Sensing of Environment 66, 1–16 (1998)
17. Anton, M., Loyola, D., Navascues, B., Valks, P.: Comparison of GOME total ozone data with ground data from the Spanish Brewer spectroradiometers. Annales Geophysicae 26, 401–412 (2008)
18. `http://weather.uwyo.edu/upperair/sounding.html`
19. Iqbal, M.: An Introduction to Solar Radiation. Academic, San Diego (1983)

A Novel Coral Reefs Optimization Algorithm for Multi-objective Problems

S. Salcedo-Sanz, A. Pastor-Sánchez,
D. Gallo-Marazuela, and A. Portilla-Figueras

Universidad de Alcalá, Alcalá de Henares, Madrid, Spain

Abstract. In this paper we detail a new algorithm for multi-objective optimization, the Multi-Objective Coral Reefs Optimization (MO-CRO) algorithm. The algorithm is based on the simulation of the coral reefs processes, including corals' reproduction and fight for the space in the reef. The adaptation to multi-objective problems is an easy process based on domination or non-domination during the process of fight for the space in the reef. The final MO-CRO is an easily implementing and fast algorithm, quite simple, but able to keep diversity in the population of corals (solutions) in a natural way. Experiments in different multi-objective benchmark problems have shown the good performance of the proposed approach in cases with limited computational resources, where we have compared it with the well known NSGA-II algorithm as reference.

1 Introduction

The Coral Reefs Optimization Algorithm (CRO) is an evolutionary bio-inspired approach based on the simulation of the processes in a coral reefs, recently proposed in [1]. The CRO can be classified into the family of bio-inspired algorithms which try to artificially simulate the behavior of a specific natural ecosystem to tackle optimization problems, similarly to ant colony optimization [2], particle swarm optimization algorithm [3], artificial bee colony approach (ABC) [4] or the weed colonization algorithm [5]. The CRO has been proven to be effective in several single-objective optimization problems, obtaining better solutions than alternative optimization algorithms in the literature.

Basically, the CRO algorithm starts from a population of individuals encoding different solutions to a given optimization problem. These solutions are located in an square grid (*reef*), where there are also empty spaces at the beginning of the algorithm. The algorithm is though to simulate the process of coral reproduction (sexual and asexual reproduction operators are applied), and the process of coral reef formation, where a fight for space occurs. Thus, in each step of the CRO algorithm a coral larvae formation is carried out, and each larva tries to occupy a place in the reef. It depends on how strong the larva is (how good the solution to the optimization problem is), or if it is lucky enough to find an empty place in the reef. Note that empty places in the reef are scarce after some generations of larvae, though a process of corals depredation ensures the possibility of empty places in the reef even at the final stages of the algorithm.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 326–333, 2013.

In this paper we describe a multi-objective version of the CRO algorithm (MO-CRO). The MO-CRO can be directly obtained from the basic CRO algorithm with very few adaptations, resulting in an effective algorithm for multi-objective optimization problems. The main characteristic of the MO-CRO is its simplicity and the easy implementation of the algorithm, and its good performance in cases where computational resources are limited. In the paper we describe in detail the algorithm and its characteristics and provide some experimental proofs of its performance in different benchmark optimization problems, where we compare with the well-known NSGA-II algorithm.

The rest of the paper has been structured as follows: next section states the main concepts and definitions of multi-objective optimization problems. Section 3 details the basic CRO algorithm, and the adaptations needed to make it appropriate to solve multi-objective optimization problems. Section 4 describes the experimental part of the paper, in which the MO-CRO has been applied to different benchmark problems and compared to the NSGA-II approach. Section 5 closes the paper by giving some final remarks and conclusions.

2 Multi-objective Problems: Basic Concepts and Definitions

Following [6], we define next several important concepts in multi-objective optimization such as domination, equivalency, non-domination with respect to sets and pareto optimal vectors and fronts, that will be helpful later on, in the definition of the MO-CRO.

Let us consider a multi-objective optimization algorithm, defined in a search space $\mathcal{S}$, formed by vectors with n-components $\mathbf{x} = x_1, \ldots, x_n$. Let us consider a vector of m objective functions $(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$, that maps each vector $\mathbf{x}$ with a fitness space. The aim of a multi-objective optimization problem is to find the set of optimal tradeoff solutions in $\mathcal{S}$ called *Pareto optimal set*. Note that usually not all the pareto set is calculated, but a representative subset is enough in the majority of cases. The following definitions are useful to clearer define the problem:

Given two parameter vectors $\mathbf{x}$ and $\mathbf{y}$, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ ($y \prec x$) iff $\mathbf{x}$ is at least as good as $\mathbf{y}$ in all objectives, and better in at least one. Similarly, $\mathbf{x}$ is equivalent to $\mathbf{y}$ ($\mathbf{x} \equiv \mathbf{y}$) iff they are identical in all objectives considered. Two parameter vectors are incomparable iff they are not equivalent, and neither dominates the other. A parameter vector $\mathbf{x}$ is non-dominated with respect to a set Ψ of vectors iff there is no vector in Ψ that dominates $\mathbf{x}$. Ψ is called *nondominated set* iff all vectors in Ψ are mutually nondominating. The set of corresponding objective vectors for a nondominated set is a nondominated front. A parameter vector is defined as *Pareto optimal* iff is nondominated with respect to the set of all parameter vectors. Note that an improvement in any one objective of a pareto optimal vector means worsening at least one other objective. The Pareto optimal set is the set of all Pareto optimal parameter vectors, and the corresponding set of objective vectors is the Pareto optimal front. Note that the Pareto optimal

set is a subset of the search space, whereas the Pareto optimal front is a subset of the fitness space.

3 The Multi-objective Coral Reefs Optimization Algorithm

3.1 Basic CRO Algorithm

The CRO is a novel meta-heuristic approach based on corals' reproduction and coral reefs formation, proposed in [1]. Basically, the CRO is based on the artificial modeling of a coral reef, Λ, consisting of a $N \times M$ square grid. We assume that each square (i, j) of Λ is able to allocate a coral (or colony of corals) $\Xi_{i,j}$, representing a solution to a given optimization problem, which is encoded as a string of numbers in a given alphabet $\mathcal{I}$. The CRO algorithm is first initialized at random by assigning some squares in Λ to be occupied by corals (i.e. solutions to the problem) and some other squares in the grid to be empty, i.e. holes in the reef where new corals can freely settle and grow in the future. The rate between free/occupied squares in Λ at the beginning of the algorithm is an important parameter of the CRO algorithm, which is denoted as ρ, and note that $0 < \rho_0 < 1$. Each coral is labeled with an associated *health* function $f(\Xi_{ij}) : \mathcal{I} \rightarrow \mathbb{R}$, that represents the problem's objective function. The CRO is based on the fact that reef will progress, as long as healthier (stronger) corals (which represent better solutions to the problem at hand) survive, while less healthy corals perish.

After the reef initialization described above, a second phase of reef formation is artificially simulated in the CRO algorithm: a simulation of the corals' reproduction in the reef is done by sequentially applying different operators. This sequential set of operators is then applied until a given stop criteria is met. Several operators to imitate corals' reproduction are defined, among them: a modeling of corals' sexual reproduction (broadcast spawning and brooding), a model of asexual reproduction (budding), and also some catastrophic events in the reef, i.e. polyps depredation. After the sexual and asexual reproduction, the set of larvae formed (new solutions to the problem), try to locate a place to grow in the reef. It could be in a free space, or in an occupied once, by fighting against the coral actually located in that place. If larvae are not successful in locate a place to grow in a given number of attempts, they are depredated in this phase.

1. *Broadcast Spawning (external sexual reproduction)*: the modeling of coral reproduction by *broadcast spawning* consists of the following steps:

   1.a. In a given step k of the reef formation phase, select uniformly at random a fraction of the existing corals ρ_k in the reef to be broadcast spawners. The fraction of broadcast spawners with respect to the overall amount of existing corals in the reef will be denoted as F_b. Corals that are not selected to be broadcast spawners (i.e. $1-F_b$) will reproduce by brooding later on, in the algorithm.

1.b. Select couples out of the pool of broadcast spawner corals in step k. Each of such couples will form a coral larva by sexual crossover, which is then released out to the water. Note that, once two corals have been selected to be the parents of a larva, they are not chosen anymore in step k (i.e. two corals are parents only once in a given step).

2. *Brooding (internal sexual reproduction)*: as previously mentioned, at each step k of the reef formation phase in the CRO algorithm, the fraction of corals that will reproduce by brooding is $1 - F_b$. The brooding modeling consists of the formation of a coral larva by means of a random mutation of the brooding-reproductive coral (self-fertilization considering hermaphrodite corals). The produced larva is then released out to the water in a similar fashion than that of the larvae generated in step 1.b.

3. *Larvae setting*: once all the larvae are formed at step k either through broadcast spawning (1.) or by brooding (2.), they will try to set and grow in the reef. First, the health function of each coral larva is computed. Second, each larva will randomly try to set in a square (i, j) of the reef. If the square is empty (free space in the reef), the coral grows therein no matter the value of its health function. By contrast, if a coral is already occupying the square at hand, the new larva will set only if its health function is better than that of the existing coral. We define a number κ of attempts for a larva to set in the reef: after κ unsuccessful tries, it will be depredated by animals in the reef.

4. *Asexual reproduction*: in the modeling of asexual reproduction (budding or fragmentation), the overall set of existing corals in the reef are sorted as a function of their level of healthiness (given by $f(\Xi_{ij})$), from which a fraction F_a duplicates itself and tries to settle in a different part of the reef by following the setting process described in Step 3. Note that a maximum number of identical corals (μ) will be allowed in the reef.

5. *Depredation in polyp phase*: corals may die during the reef formation phase of the CRO algorithm. At the end of each reproduction step k, a small number of corals in the reef can be depredated, thus liberating space in the reef for next coral generation. The depredation operator is applied with a very small probability P_d at each step k, and exclusively to a fraction F_d of the worse health corals in Λ.

3.2 Multi-objective CRO

The adaptation of the CRO to multi-objective problems is an easy task by starting from the basic CRO approach. It is established in the *Larvae Setting* process of the algorithm and in a generalization of the depredation operator: once all the larvae from broadcast spawning and brooding have been produced, they start the setting process one by one, trying to established themselves into the reef. When an existing coral occupies a given position in the reef that is tried by a

larva, a fight for the space occurs. In the MO-CRO, this fight for the space is based on domination of solutions. Let us call Ξ_A to the coral currently occupying a given location on the reef, and Ξ_B the larva challenging for the space. In the MO-CRO, Ξ_B wins the fight (and occupies the place of Ξ_A) iff $\Xi_A \prec \Xi_B$. In any other case, Ξ_A wins, and the challenging larva either tries another place in the reef, or die, depending on its current κ value. Note that in case of equivalency between solutions ($\Xi_A \equiv \Xi_B$), the current solution in the reef is maintained. A second adaptation is needed in order to provide diversity to the reef: In the MO-CRO, a fix number μ of corals with the same value in all objective functions is allowed in the population. After the larvae setting process, we obtain the number of corals in the reef with the same value in all objectives, let us call it β, and if $\mu < \beta$, $(\beta - \mu)$ randomly chosen corals are depredated. We call this adaptation as Extreme Depredation Operator (EDO).
Several points must be notated:

1. The algorithm will never destroy vectors belonging to the pareto optimal front, since they will dominate or will be incomparable to any other possible vector in the search space.
2. Dominated solutions may coexist with better individuals if empty places exist in the coral reef.
3. The generation of diversity in the first stages of the algorithm is comparable to that of evolutionary approaches.
4. Due to 1., the genetic drift causing the loss of diversity in the population of evolutionary algorithms does not occur within the CRO algorithm.
5. Thanks to the EDO, the The CRO does not, therefore, need to establish a ranking on the individuals, nor needs for modification in fitness values in order to preserve diversity in the population.
6. The MO-CRO is nearly as fast as the basic CRO, and it is straightforward to implement from the basic CRO version.

4 Experimental Part

In order to test the performance of the proposed MO-CRO, we have tackled a number of benchmark functions, usually employed in the literature for the evaluation of multi-objective approaches [7], that are defined in Table 1. In order to contrast the results obtained with the MO-CRO, we will carry out a comparison with a successful algorithm in the literature, the NSGA-II algorithm [7]. The methodology of the comparison will be the following: a fix (limited) number of function evaluations is set for both the NSGA-II and MO-CRO. The idea is to analyze the performance of the algorithms in cases of reduced computational resources available. Specifically, the NSGA-II has been run with 100 individuals, during 200 generations, and the MO-CRO has been adjusted to approximately evaluate the same number of functions. A matlab implementation of the NSGA-II algorithm has been used [8]. In this version of the algorithm, a Simulated Binary Crossover [9] and a polynomial mutation [10] are implemented. We have also used this operators in the MO-CRO algorithm.

Table 1. Benchmark function to test the MO-CRO proposed in this paper

Function	n	variable bounds	Expression
FON	3	$[-4, 4]$	$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_1^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$
			$f_2(\mathbf{x}) = 1 - \exp\left(-\sum_1^3 \left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$
SCH	1	$[-1000, 1000]$	$f_1(x) = x^2$
			$f_2(x) = (x-2)^2$
ZDT1	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$
			$f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{\frac{x_1}{g(\mathbf{x})}}\right]$
			$g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1)$
ZDT2	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$
			$f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - (\frac{x_1}{g(\mathbf{x})})^2\right]$
			$g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1)$
ZDT3	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$
			$f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} - \frac{x_1}{g(\mathbf{x})}\sin(10\pi x_1)\right]$
			$g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1)$
ZDT6	10	$[0, 1]$	$f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$
			$f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - (\frac{f_1(\mathbf{x})}{g(\mathbf{x})})^2\right]$
			$g(\mathbf{x}) = 1 + 9\left[\left(\sum_{i=2}^n x_i\right)/(n-1)\right]^{0.25}$

The comparative results obtained with the MO-CRO and NSGA-II algorithms are shown in Figure 1. As can be seen, in the FON function, the MO-CRO is able to get the optimal pareto front, though in this case the NSGA-II algorithm covers better the extremes of the front. In SCH benchmark the MO-CRO covers the optimal front in a similar way that the NSGA-II. In the most difficult functions (ZDT), the performance of the MO-CRO with the resources considered (number of function evaluation) is better than the NSGA-II. In ZDT1 function, the NSGA-II does not obtain the optimal pareto front with the number of function evaluations fixed. The MO-CRO is able to get this optimal front, covering it in a reasonable way. The behavior of the algorithms is quite similar in function ZDT2, the NSGA-II is not able to obtain the optimal pareto front with the limited number of objective function evaluation, and the MO-CRO obtains it, reasonably covering the complete front. Function ZDT3 considers a discontinuous optimal front. The MO-CRO is able to obtain it, but it does not cover completely part of the front. The NSGA-II obtains a suboptimal front, but it covers it almost completely. Finally, in Function ZDT6, both algorithms as able to obtain the optimal pareto front. The NSGA-II covers it better than the MO-CRO, but it also shows suboptimal solutions at the left-most part of the front.

Summarizing, the proposed MO-CRO obtains good results in terms of optimal front location even with limited computational resources (limited number of possible function evaluation), improving the behaviour of the NSGA-II algorithm. It seems that the NSGA-II outperforms the MO-CRO in terms of diversity of solutions in the optimal front. Due to its simplicity, the proposed MO-CRO algorithm could be a good option in real-world multi-objective problems and applications.

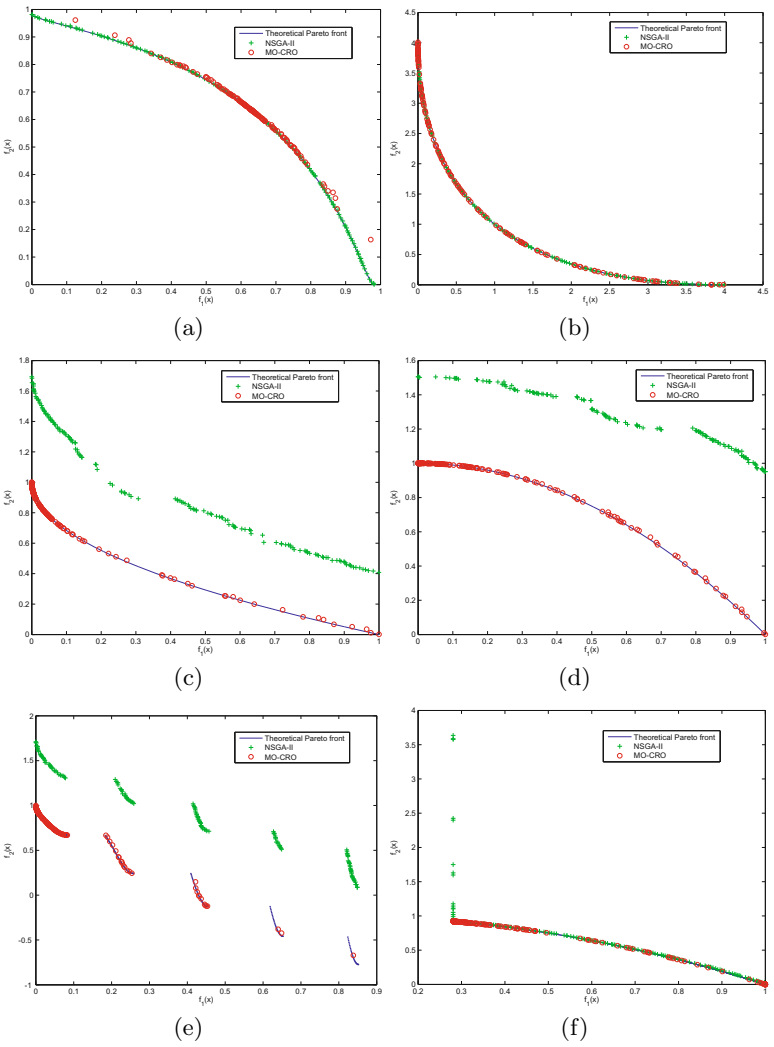


Fig. 1. Comparison MO-CRO vs. NSGA-II in the different benchmark functions considered; (a) FON; (b) SCH; (c) ZDT1; (d) ZDT2; (e) ZDT3 and (f) ZDT6.

5 Conclusions

In this paper we have presented a novel algorithm for multi-objective optimization problems: the Multi-Objective Coral Reefs Optimization algorithm (MO-CRO). We have detailed the algorithm's structure and the adaptations necessary in the CRO to convert it into a multi-objective algorithm. We have verified the goodness of the proposed approach in several multi-objective optimization benchmark problems with limited computational resources available, where we have compared its performance with that of a popular evolutionary algorithm for multi-objective optimization (the NSGA-II algorithm). The proposed MO-CRO main characteristics are its simplicity and easy implementation, its low-computational cost and the good results obtained in several test functions.

Acknowledgements. This work has been partially supported by Spanish Ministry of Science and Innovation, under project numbers ECO2010-22065-C03-02.

References

1. Salcedo-Sanz, S., Del Ser, J., Gil-López, S., Landa-Torres, I., Portilla-Figueras, J.A.: The Coral Reefs Optimization Algorithm: A new metaheuristic algorithm for hard optimization problems. In: Proc. of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA), Mataró, Barcelona (2013)
2. Dorigo, M., Maziezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating ants. IEEE Transactions on Systems, Man and Cybernetics B 26(1), 29–41 (1996)
3. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of the 4th IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
4. Karaboga, D., Basturk, B.: On the performance of the artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)
5. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1, 355–366 (2006)
6. Huban, S., Hingston, P., Barone, L., While, L.: A Review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
7. Deb, K., Pratab, A., Agrawal, S., Merayivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
8. `http://www.mathworks.es/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm`
9. Deb, K., Agarwal, R.B.: Simulated Binary Crossover for continuous search space. Complex Systems 9, 115–148 (1995)
10. Raghuwanshi, M.M., Kakde, O.G.: Survey on multiobjective evolutionary and real coded genetic algorithms. In: Proc. of the 8th Asia Paciffc Symposium on Intelligent and Evolutionary Systems, pp. 150–161 (2004)

Fuzzy Clustering with Grouping Genetic Algorithms

S. Salcedo-Sanz[1], L. Carro-Calvo[1], A. Portilla-Figueras[1], L. Cuadra[1], and D. Camacho[2]

[1] Universidad de Alcalá, Madrid, Spain
sancho.salcedo@uah.es
[2] Universidad Autónoma de Madrid, Madrid, Spain

Abstract. This paper presents a novel approach to fuzzy clustering based on Grouping Genetic Algorithms (GGAs). Our approach consists of a GGA devised for fuzzy clustering by means of a novel encoding of individuals (containing elements and clusters sections), a new fitness function (a superior modification of the Davies-Boudin index) and specially tailored crossover and mutation operators. The overall performance of our approach has been tested in a variety of fuzzy clustering problems, showing very good performance in all cases.

1 Introduction

Fuzzy clustering stands for the class of clustering problems where the boundary between clusters is ill-defined, in the sense that a given sample is allowed to belong to different clusters [1]. As such, the notion of *fuzziness* becomes relevant since any object of the data set is assigned to a given cluster with some *membership grade*, usually set between 0 and 1 (low and high membership grade, respectively). Formally, if $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is a set of N data vectors in a given observation space $\mathcal{S}$, the goal of a fuzzy clustering algorithm is to find a partition of X in a finite number of k clusters, so that a data vector $\mathbf{x}_j$ can belongs to a cluster $C_i \in \{C_1, \ldots, C_k\}$ with a degree of membership $u_{ij} \in [0, 1]$. Let $\boldsymbol{\mu}_i$ be the centroid of those objects that result to be grouped in the cluster C_i. This is equivalent to find a "partition matrix" $\mathbf{U}$ whose elements $u_{ij} \in [0, 1]$ (with $1 \leq i \leq k$ and $1 \leq j \leq N$) fulfill the conditions

$$\sum_{i=1}^{k} u_{ij} = 1, \ \forall j \in \{1, \ldots, N\}, \tag{1}$$

$$0 < \sum_{j=1}^{N} u_{ij} < N, \ \forall i \in \{1, \ldots, k\}. \tag{2}$$

Meta-heuristic algorithms have been thoroughly applied to fuzzy clustering in the last years, due to their superior properties of robustness and convergence to near-optimal solutions at a moderate computational cost. Many of these approaches are based on evolutionary variants of the C-means algorithm [2–4]. In

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 334–341, 2013.

this paper we proposes a grouping genetic algorithm for fuzzy clustering. The grouping genetic algorithm (GGA) [5, 6] is a class of evolutionary algorithms whose encoding procedure is especially designed to deal with grouping problems. It has been successfully applied to a variety of problems involving grouping of items. However, to the best of the authors' knowledge, its performance has not been tested yet in fuzzy clustering problems. On this purpose, and as will be explained throughout this paper, we present a novel grouping encoding, a modified objective function, and different operators (i.e. modified crossover and mutation) specifically adapted to the fuzzy clustering problem and thus, aiming at improving the performance of GGA when applied to this paradigm. Simulation results are presented so as to assess the performance of the proposed scheme in a number of application scenarios, based on which it is concluded that the GGA-based procedure here presented outperforms conventional Fuzzy C-Means clustering methods.

The rest of this paper is structured as follows: next section summarizes some fuzzy clustering evaluation measures. Section 3 presents the GGA approach proposed in this paper and Section 4, shows the performance of the proposed approach in a variety of different synthetic fuzzy clustering problems. Finally, Section 5 completes the paper by discussing some concluding remarks.

2 Fuzzy Clustering Evaluation

The evaluation of a given solution in a fuzzy clustering problem can be carried out using two different strategies:

2.1 Unsupervised Evaluation

For comparison purposes with the objective (fitness) function later proposed for evaluating the performance of the algorithm, we use the following unsupervised measures in the related literature:

- Xie-Beni index (XB): defined in [7], this measure combines the sum of square errors with a term for measuring clusters separation,

$$XB(\mathbf{U}) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{N} u_{ij}^{\alpha} \cdot d^2(\mathbf{x}_j, \boldsymbol{\mu}_i)}{N \cdot \min\limits_{1 \le i,j \le k, i \ne j} \left\{ d^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \right\}}. \tag{3}$$

- Fukuyama-Sugeno index (FS): the FS index [8] is similar to the XB index but, in this case, the separation between clusters is evaluated with respect to the average centroid of the k clusters, $\boldsymbol{\mu}^* = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\mu}_i$, instead of the centroid of the rest of clusters. Based on this rationale we obtain

$$FS(\mathbf{U}) = \sum_{i=1}^{k}\sum_{j=1}^{N} u_{ij}^{\alpha} \cdot d^2(\mathbf{x}, \boldsymbol{\mu}_i) - \sum_{i=1}^{k}\sum_{j=1}^{N} u_{ij}^{\alpha} \cdot d^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}^*). \tag{4}$$

The aforementioned unsupervised measures are useful in those problems in which there is no additional information to check the quality of the generated clusters. However, there are some clustering problems in which such information is indeed available, hence allowing for supervised measures.

2.2 Supervised Measurement: Rand Index

Among the supervised measures – sometimes called *external* measures – in this work the well-known Rand index (R) will be utilized after defuzzification of the samples. It computes the similarity between the obtained partition and the *known* optimal solution as follows,

$$R(\mathbf{U}) = \frac{TP + FN}{TP + FP + TN + FN} \tag{5}$$

where TP and FP are the number of correct and incorrect assignments, respectively, when the decision consists in assigning two elements to the *same* cluster; and TN and FN are the number in correct and incorrect assignments, respectively, when the decision consists of assigning two elements to *different* clusters. In words, it is a measure of the percentage of correct decisions taken by the algorithm. Note that the value of R lies on the interval $[0, 1]$: values of R closer to 1 indicate a better quality of the solution tested.

3 Proposed Grouping Genetic Algorithm for Fuzzy Clustering

As mentioned in Section 1, a GGA, the encoding procedure, crossover and mutation operators of traditional GAs are modified to yield a compact algorithm, with improved performance in grouping-based problems. We will detail these parts of the algorithm in the next subsections.

3.1 Problem Encoding

The proposed GGA for fuzzy clustering is a variable-length genetic algorithm, with a novel encoding to deal with this specific problem. The encoding is carried out by splitting each chromosome in the algorithm (or equivalently, its corresponding individual or candidate solution) into two parts: $\mathbf{c} = [\mathbf{U}|\mathbf{g}]$. The first part is the *element* section composed by the partition matrix $\mathbf{U}$, whereas the second part is denoted as the *group* section of the individual. Following this notation, a certain individual for a fuzzy clustering problem with N elements (objects or observations) and k clusters can be expressed as

$$\left[\begin{array}{ccc|c} u_{1,1}, & \ldots, & u_{1,N} & \\ \vdots, & \ldots, & \vdots & g_1, g_2, \ldots, g_k \\ u_{k,1}, & \ldots, & u_{k,N} & \end{array}\right],$$

where it is important to note that each element $u_{i,j}$ represents the degree of membership of j-th observation to i-th cluster, whereas the group section keeps a list of tags associated to each of the clusters of the solution. Also observe that in this encoding, not only the group section has a variable length, but also the element section, since the number of clusters is also a variable of the problem. The initialization of the population is carried out at random in the first step of the algorithm. The number of clusters are obtained from a uniform distribution in the range $[2, C_{max}]$, where C_{max} is a parameter of the specific problem being solved. After setting the number of cluster for each individual, the u_{ij} variables are randomly chosen in the interval $[0, 1]$.

3.2 Objective Function

In this paper we also propose an adaptation of the well-known Davis-Bouldin index (used in crisp clustering problems) to the fuzzy case which, to the best of our knowledge, is novel in fuzzy clustering. We will show that the use of this modified index renders better results for the GGA than the other existing evaluation indices. The idea of the Davis-Boudin index for crisp clustering problems is to minimize the intra-cluster distances while simultaneously maximizing the distances among the different clusters, yielding

$$DB(\mathbf{U}) = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left\{\frac{\sum\limits_{x\in C_i} d^2(\mathbf{x},\boldsymbol{\mu}_i) + \sum\limits_{x\in C_j} d^2(\mathbf{x},\boldsymbol{\mu}_j)}{d^2(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)}\right\}. \tag{6}$$

In the above expression note that small values of the conventional DB index corresponds to compact and well-separated clusters. The adaptation of the DB index for fuzzy clustering proposed in this work is expressed as

$$MDB(\mathbf{U}, d) = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left\{\frac{\sum\limits_{t=1}^{N} u_{i,t}^{\alpha} d^2(\mathbf{x}_t,\boldsymbol{\mu}_i) + \sum\limits_{t=1}^{N} u_{j,t}^{\alpha} d^2(\mathbf{x}_t,\boldsymbol{\mu}_j)}{d^2(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)}\right\}, \tag{7}$$

where μ_i stands for the centroid associated to cluster C_i, calculated by considering the average of each observation weighted by the degree of membership to cluster C_i. Note in Expression (7) that the proposed MDB index explicitly depends on the particular definition considered for the distance d. For example, if we consider the GK distance and based on the covariance matrices of the clusters, the DB index for fuzzy clustering problems will be given by

$$MDB(\mathbf{U}, d_{GK}) = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left\{\frac{\sum\limits_{t=1}^{N} u_{i,t}^{\alpha} d^2_{|\Sigma_i|^{1/d}\Sigma_i^{-1}}(\mathbf{x}_t,\boldsymbol{\mu}_i) + \sum\limits_{t=1}^{N} u_{j,t}^{\alpha} d^2_{|\Sigma_j|^{1/d}\Sigma_j^{-1}}(\mathbf{x}_t,\boldsymbol{\mu}_j)}{\min\left\{d^2_{|\Sigma_i|^{1/d}\Sigma_i^{-1}}(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j), d^2_{|\Sigma_j|^{1/d}\Sigma_j^{-1}}(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)\right\}}\right\}. \tag{8}$$

3.3 Crossover Operator

The crossover operator implemented in the grouping genetic algorithm used in this paper is a modified version of the one initially proposed by Falkenauer [5], but with the added bonus of being adapted to the fuzzy clustering problem. These are the main steps followed to complete the crossover operation: 1.) Select two individuals at random, and choose two crossing points in their group part; 2.) Insert the elements belonging to the selected groups of the first individual into the offspring; 3.) Assign the degree of membership of the inserted elements equal to the first individual; 4.) Insert the elements belonging to the selected groups of the second individual into the offspring; 5.) Assign the degree of membership of the inserted elements, in the following way: first, the remaining degree membership after the assignment of the elements of the first individual is calculated. This remaining degree membership is then proportionally shared among the elements of the second individual; 6.) Remove empty clusters, if any; 7.) Modify the labels of the current groups in the offspring in order to numerate them from 1 to k.

3.4 Mutation Operator

Mutation operators include modifications in each individual of the population with a low probability, in order to explore new regions of the search space, and also scape from local optima when the algorithm is near convergence. In this case, we have implemented two different mutation operators adapted to the fuzzy clustering problems, first a mutation by *cluster splitting*, that consists of splitting a selected cluster in two different parts and the degrees of membership are also randomly split between the two new clusters. And second a mutation by *clusters merging*, that consists of randomly selecting two existing clusters and merging them into just one single cluster. The degree of membership of the new cluster is the sum of the degrees of the previous ones.

4 Experiments and Results

This section summarizes and discusses the experimental work we have carried out in order to assess the performance of our proposed GGA approach. We have explored a number of variations of the proposed GGA in different synthetic fuzzy clustering problems, and also compared it to a Fuzzy C-Means (FCM) algorithm [1].

Experiment 1: Spherical Clusters. In this first experiment, we test the performance of the proposed GGA in a two-dimensional clustering problem, defined by 300 observations randomly generated using a Gaussian distribution from 8 equiprobable classes, with mean values $\mu_1 = (-1, 1)$, $\mu_2 = (2, -2)$, $\mu_3 = (1, 0)$, $\mu_4 = (3, -1)$, $\mu_5 = (-1, -1)$, $\mu_6 = (-1, -3)$, $\mu_7 = (1, 2)$, $\mu_8 = (3, 1)$, and covariance matrices:

$$\Sigma_1 = \Sigma_2 = \ldots = \Sigma_8 = \begin{bmatrix} 0.35^2 & 0 \\ 0 & 0.35^2 \end{bmatrix}. \tag{9}$$

Table 1. Comparison of the results (in terms of the number of clusters finally found, and as a function of the Rand index) obtained by the proposed GGA algorithm with MDB, XB and FS indexes, respectively, and the FCM algorithm in the first synthetic clustering problem considered.

Algorithm	# clusters	Rand Index
GGA (MDB index)	8	**0.9918**
GGA (XB index)	8	0.9785
GGA (FS index)	9	0.9847
FCM	7	0.9512

Note that this procedure results in a problem characterized by spherical clusters.

We have applied to this problem a number of configurations of the proposed GGA – with MDB, XB and FS objective (fitness) functions – and the FCM algorithm. In this regard, Table 1 lists the supervised evaluation of the results obtained by the aforementioned algorithms. Note that the proposed GGA with the three different objective functions obtains better results than the FCM algorithm. In particular, the GGA with our MBD index exhibits the best behavior ($R = 0.9918$), higher than that of the conventional FCM algorithm ($R = 0.9512$). In addition, note that the GGA with DB and XB indexes achieves the solution with the optimal number of clusters (i.e. 8). In order to better describe the behavior of the best algorithm Figures 1 shows the two-dimensional distributions of the 8 clusters found. The color of each observation has been obtained as a combination of those colors representing each cluster, weighted by the degree of membership of each observation.

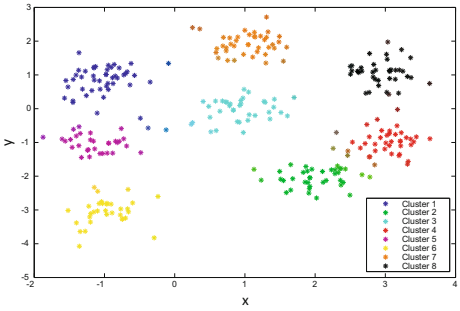


Fig. 1. Representation of the best result obtained by the proposed GGA with MDB fitness function in the first clustering example

Experiment 2: Unbalanced Data. We now test the performance of the proposed GGA in a different two-dimensional clustering problem, defined by 400 randomly generated objects following a distribution drawn from 3 Gaussian classes with probabilities $p_1 = 0.5$, $p_2 = 0.33$ and $p_3 = 0.17$. The mean values of each of such classes are $\mu_1 = (0, 2)$, $\mu_2 = (-1, -1)$, $\mu_3 = (2, -1)$, whereas their covariance matrices are given by

$$\Sigma_1 = \begin{bmatrix} 1^2 & 0 \\ 0 & 0.8^2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.6^2 & 0 \\ 0 & 0.4^2 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 0.3^2 & 0 \\ 0 & 0.5^2 \end{bmatrix}, \quad (10)$$

Note that, in this case, the classes are not spherical, and have different distributions.

Table 2. Comparison of the results (in terms of the number of clusters finally found, and as a function of the Rand index) obtained by the proposed GGA algorithm with MDB, XB and FS indexes, respectively, and the FCM algorithm in the second synthetic clustering problem considered.

Algorithm	# clusters	Rand Index
GGA (MDB index)	3	**0.9177**
GGA (XB index)	3	0.9128
GGA (FS index)	7	0.7606
FCM	4	0.8561

Table 2 shows, in terms of the Rand index, the results obtained by the proposed GGA with DB, XB and FS index, and those achieved by the FCM algorithm. As shown in this Table, the GGA with MDB and XB indexes obtain similar results (better than the FCM), whereas the result of the GGA with FS index is slightly worse than the result of the FCM algorithm. The best results correspond to the GGA algorithm with the MDB index, rendering a value of $R = 0.9177$ (higher than that of the FCM algorithm), and what is very important, finding the 3 clusters hidden in the data. Finally, Figure 2 illustrates, in a more intuitive way, the fuzzy clustering reached by the proposed GGA using the MDB index as objective function.

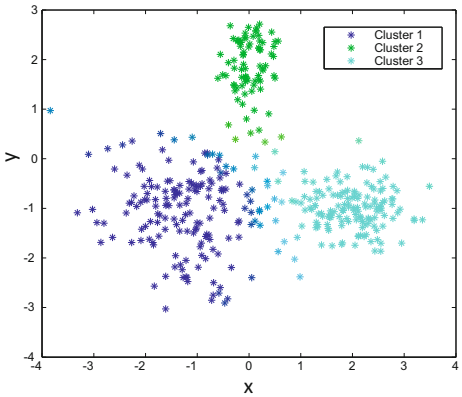


Fig. 2. Best result obtained by the proposed GGA (MDB index) in the second clustering example considered.

5 Conclusions

In this paper we have presented a grouping genetic algorithm for fuzzy clustering problems. The main contributions of this work are 1) a novel encoding approach of the individuals involved in the evolutionary process, containing information not only of the partition matrix elements, but also of the clusters being obtained; 2) a novel fitness function based on a modification of the Davis-Boudin index for its efficient use in fuzzy clustering problems, and that enables the chance of introducing norms adapted to any problem and 3) novel crossover and mutation operators particularly derived to achieve the effective evolution of the individuals.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Economy, under project grants ECO2010-22065-C03-02 and TIN2010-19872/TSI.

References

1. Baraldi, A., Blonda, P.: A survey of fuzzy clustering algorithms for pattern recognition, part I. IEEE Trans. on Systems, Man and Cybernetics, part B 29, 778–785 (1999)
2. Hwang, C., Rhee, F.C.: Uncertain fuzzy clustering: interval type-2 fuzzy approach to C-means. IEEE Trans. on Fuzzy Systems 15, 107–120 (2007)
3. Gan, G., Wu, J., Yang, Z.: A genetic fuzzy-Modes algorithm for clustering categorical data. Expert Systems with Applications 36(2), 1615–1620 (2009)
4. Maulik, U., Saha, I.: Automatic fuzzy clustering using modified differential evolution for image classification. IEEE Trans. on Geoscience and Remote Sensing 48(9), 3503–3510 (2010)
5. Falkenauer, E.: Genetic algorithms for grouping problems. Wiley, New York (1998)
6. Agustín-Blas, L.E., Salcedo-Sanz, S., Jiménez-Fernández, S., Carro-Calvo, L., Del Ser, J., Portilla-Figueras, J.A.: A new grouping genetic algorithm for clustering problems. Expert Systems with Applicacions 39, 9695–9703 (2012)
7. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 13, 841–847 (1991)
8. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy C-means method. In: 5th Fuzzy Systems Symp., pp. 247–250 (1989)

Graph-Based Substructure Pattern Mining Using CUDA Dynamic Parallelism

Fei Wang, Jianqiang Dong, and Bo Yuan

Intelligent Computing Lab, Division of Informatics
Graduate School at Shenzhen, Tsinghua University
Shenzhen 518055, P.R. China
wangfeifast@gmail.com,513712287@qq.com,yuanb@sz.tsinghua.edu.cn

Abstract. CUDA is an advanced massively parallel computing platform that can provide high performance computing power at much more affordable cost. In this paper, we present a parallel graph-based substructure pattern mining algorithm using CUDA Dynamic Parallelism. The key contribution is a parallel solution to traversing the DFS (Depth First Search) code tree. Furthermore, we implement a parallel frequent subgraph mining algorithm based on the subgraph mining techniques used in gSpan and the entire subgraph mining procedure is executed on GPU to ensure high efficiency. This parallel gSpan is functionally identical to the original gSpan and experiment results show that, with the latest CUDA Dynamic Parallelism techniques, significant speedups can be achieved on benchmark datasets, particularly in traversing a DFS code tree.

Keywords: DFS, gSpan, GPU, CUDA, Dynamic Parallelism.

1 Introduction

Graph-based substructure pattern algorithms have been widely applied in many research areas such as molecular substructure discovery, bioinformatics pattern mining [14], network link analysis [12], social network data [13] and financial data analysis [9]. Since the introduction of frequent pattern mining in 1990s [1], many subgraph mining algorithms have been developed such as AGM [6], FSG [8], gSpan [17] and Gaston [10, 11]. Previous research has shown that gSpan is one of the best algorithms in terms of running time and memory cost [16]. To the best of our knowledge, there was no GPU (Graphics Processing Unit) accelerated gSpan, although a parallel gSpan using SGI Altix 3000 is available [2].

A major challenge in subgraph mining is the subgraph isomorphism test (to check whether two graphs have the same structure), which is an NP-complete problem [17]. In gSpan, a novel graph-labeling mechanism called DFS Lexicographic Order is employed to reduce the search space significantly. With the minimum DFS code, it is possible to efficiently determine whether an isomorphism exists between two graphs [17]. In the meantime, unlike previous algorithms, gSpan only produces candidate graphs existing in the dataset.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 342–349, 2013.

However, the search space of gSpan is represented by a DFS code tree and the time complexity is very high to traverse a DFS code tree. Furthermore, DFS is an inherently sequential process [4], making the parallel design of gSpan problematic due to the heavily required threads synchronization and communication. Luckily, with the emergence of the groundbreaking GPU computing technology, especially the introduction of NVIDIA's CUDA (Compute Unified Device Architecture) platform in 2007, the domain of high performance computing is undergoing a major revolution. CUDA is an abstract heterogeneous parallel programming model for researchers in various disciplines to make full use of the power of GPUs conveniently.

The rest of the paper is organized as follows. Section 2 introduces the basic concepts of gSpan. Section 3 explains the key techniques of CUDA. A parallel tree structure for DFS is detailed in Section 4. Section 5 presents the framework of the parallel gSpan algorithm. Experiment results are shown in Section 6 and this paper is concluded in Section 7 with some directions for future research.

2 Foundations of gSpan

A key feature of gSpan is to use the DFS code to label a graph and test the isomorphism of subgraphs by minimum DFS codes. The idea of gSpan is to label a subgraph with a DFS code and produce child DFS codes from the right-most path of the DFS tree. If the child DFS code is a minimum DFS code, the corresponding graph is processed. This process is executed recursively until the child DFS code is not a minimum one.

Definition 1 (*DFS code*)
A DFS tree T is built through the DFS of a graph G. The depth-first traversal of the vertices forms a linear order, which can be used to label these vertices. An edge sequence (e_i) is produced as follows.

Assume $e_1 = (i_1, j_1)$ and $e_2 = (i_2, j_2)$: (1) if $i_1 = i_2$ and $j_1 < j_2$, then $e_1 < e_2$; (2) if $i_1 < i_2$ and $j_1 = j_2$, then $e_1 < e_2$; (3) if $e_1 < e_2$ and $e_2 < e_3$, then $e_1 < e_3$. The sequence e_i where $i = 0, \ldots, |E|-1$ is called a DFS code, denoted as *code* (G,T).

Definition 2 (*DFS Lexicographic Order*)
Suppose Z = {*code* (G, T)| T is a DFS tree of G}, Z is a set containing all DFS codes for all the graphs. DFS Lexicographic Order is a linear order of DFS codes defined as follows.

If α = *code* $(G_\alpha, T_\alpha) = (a_0, a_1, \ldots, a_m)$ and β = *code* $(G_\beta, T_\beta) = (b_0, b_1, \ldots, b_n)$, $\alpha, \beta \in Z$, if and only if one of the two conditions are true, $\alpha \leq \beta$ is true:

(1) $a_k = b_k$ for $0 \leq k \leq m$ and $n \geq m$; (2) $\exists t$, $0 \leq t \leq min\ (m, n)$, $a_k = b_k$ for $k < t$, $a_t < b_t$.

Definition 3 (*Minimum DFS Code*)
In the set Z (G), according to the DFS Lexicographic Order, the minimum one is called Minimum DFS Code of G, denoted as $min\{Z(G)\}$.
Theorem 1: G and G' are isomorphic, if and only if $min\{Z(G)\} = min\{Z(G')\}$.

Definition 4 (*Rightmost Path Extension Rules*)
Given a DFS code s and an edge e, in either of the following two cases, $s \cup e$ is called the rightmost path extension: (1) e connects the rightmost vertex and the vertices on the rightmost path in the DFS tree; (2) e connects the rightmost path in the DFS tree and a new vertex.

Fig. 1 shows an example of the DFS Code Tree search space. The details of the sequential version of gSpan are shown in *Algorithm* 1 and *Subprocedure* 1.

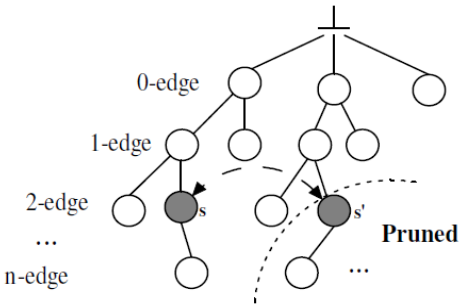


Fig. 1. DFS code tree search space [17]

```
Algorithm 1: GraphSet_Projection(D,S)
1: sort the labels in D by their frequency;
2: remove infrequent vertices and edges;
3: relabel the remaining vertices and edges;
4: S^1 ← all frequent 1-edge graphs in D;
5: sort S^1 in DFS lexicographic order;
6: S ← S^1;
7: for each edge e ∈ S^1 do
8: initialize s with e, set s.D by graphs containing e;
9: Subgraph_Mining(D,S,s);
10:D ← D - e;
11:if |D|<minSup;
12:break; end if;
13:end for;
```

```
Subprocedure 1: Subgraph_Mining(D,S)
1: if s ≠ min(s)
2: return; end if;
3: S ← S ∪{s};
4: enumerate s in each graph in D and count its children;
5: for each c that is a child of s do
5: if support(c) ≥ minSup
6: s ← c;
7: Subgraph_Mining(D,S,s);
8: end if; end for;
```

3 Key CUDA Techniques

3.1 CUDA Programming Model

CUDA is a hierarchical and heterogeneous programming architecture for NVIDIA GPUs, which adopts the SIMT (Single Instruction, Multiple Thread) execution model and provides an interface for CPUs and GPUs to work collaboratively. On the hardware side, a GPU contains a number of SMs (Stream Multiprocessor) each of which consists of a group of SPs (Streaming Processor). From the programming perspective, threads are organized in to blocks, which are further grouped into a grid. A thread block is the basic execution unit for a kernel function, which is a section of code to be run on the GPU. The execution of a kernel function amounts to executing potentially a large number of threads in different blocks in parallel. Blocks can be assigned to different SMs depending on the availability but only threads in the same block can communicate with each other. The organization strategy of threads can reduce the management overhead and provide transparent scalability for CUDA programs by automatically exploiting the ever increasing number of CUDA cores in new generations of GPUs.

3.2 Concurrent Kernel Execution

This technique was first introduced with the Fermi architecture in 2010. It allows different kernels of the same application context to be executed on the GPU at the same time, making it possible for a number of small kernels to share the whole GPU [15]. With this technique, we can execute a lot of jobs in parallel without concerning the load balance problem. For example, we can start many kernels to generate children DFS codes concurrently.

3.3 CUDA Dynamic Parallelism

CUDA Dynamic Parallelism refers to the capability of a GPU to generate new work for itself [7], without the involvement of the CPU. As described in Section 1, traversing the DFS code tree in a parallel manner has long been an issue and in this paper we will show that, with this advanced capability, a tree structure can be effectively implemented in parallel.

3.4 Atomic Operations

An atomic function performs a read-modify-write atomic operation on a single 32-bit or 64-bit word residing in the global or shared memory [3]. The operation is atomic in the sense that it is guaranteed to be performed without interference from other threads [3]. Therefore, atomic operation is an important method to synchronize parallel processes and prevent race conditions. In GK110, atomic operation throughput to a common global memory address has been improved by 9 times, compared to the Fermi architecture [7]. In our implementation, it is used to maintain the child arrays when generating candidate subgraphs in parallel.

4 The Parallel DFS Code Tree

As described in Section 2, the search space of gSpan is a DFS code tree and the time complexity to search the DFS code Tree of a graph is very high. In our parallel model, the runtime of this procedure can be greatly reduced. The main idea is to generate the child nodes in parallel under the condition that there is no communication among threads. To achieve this goal, a Dynamic Parallel Tree Model is proposed (*Algorithm* 2) and an example is shown in Fig. 2.

In this model, three variables are used to trace the path. The first one is a stack, which is accessible by all threads. It is designed to record the traversed path to implement the backtracking in DFS, which is similar to the traditional DFS stack except that the structure of the stack is a tree. The second variable *g2s* is a one-dimensional array used to record the visiting order of the nodes. The third variable *f* is a two-dimensional array used to indicate whether an edge has been visited. The latter two variables are private to a DFS code node and they are both passed to the next generation in parallel.

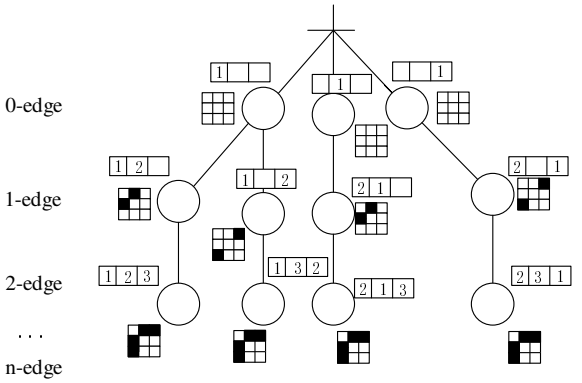


Fig. 2. Traversing the DFS code tree in parallel

```
Algortithm 2: Traverse_DFS_Code_Tree
1: Kernel Traverse(g,stack,f,g2s,depth,next) //next is nodeID
2: if(depth ≥ s.size)return; end if;
3: __shared__ int flag; //pop_stack flag shared by all threads
4: while(stack!=Null)
5: flag=true;
6: x=stack_top; y=g.edge_next[x][tid] //tid is threadID
7: if(f[x][y]==false) //if the edge hasn't been visited
8: flag=true;
9: if(g2s[y]<0)   //if the node hasn't been visited
10: push(y); copy f,g2s to child nodes;
11: child_g2s[y]=next;
```

```
12: child_f[x][y]=true; child_f[y][x]=true;
13: Traverse(g,stack,child_f,child_g2s,depth+1,next+1);
14: else
15: copy f,g2s to child nodes;
16: child_f[x][y]=true; child_f[y][x]=true;
17: Traverse(g,stack,child_f,child_g2s,depth+1,next);
18: end if;
19: end if;
20: if(flag)break; end if;
21: pop_stack;
22: end_while; end Kernel;
```

5 Parallel gSpan Implementation

In gSpan, *Subprocedure* 1 takes more than 90% of the computing time. As a result, the entire procedure is executed on the GPU while the CPU is mainly responsible for data initialization and the scheduling job in *Algorithm* 1. There are two expensive components in *Subprocedure* 1: determining whether a DFS code is a minimum one (thread 0 in *Algorithm* 3) and, if necessary, generating its child DFS codes in parallel. When the two components are finished, kernel functions are concurrently executed to process the new child DFS codes.

```
Algorithm 3: Parallel_Subgraph_Mining(D,S,s)
1: if(tid==0)  //tid for threadId
2: Kernel_BuildGraph(g,s);
3: Kernel_Travel_Is_mini(g,s,is_mini);
4: if(!is_mini)return; end if;
5: S ← S ∪ {s};
6: end if;
7: Kernel_GenerateChildren(D[s.D[tid],s,c[tid]]);
8: if support(c[tid]) ≥ mini_support
9: copy s to child_s[tid] and child_s[tid] ← c[tid];
10:Parallel_Subgraph_Mining(D,S,child_s[tid]);
11:end if;
```

6 Experiments

To evaluate the performance of our parallel gSpan algorithm, we conducted two sets of experiments to test the parallel traversing algorithm and the parallel gSpan algorithm with the same dataset[1] used in [17]. The dataset contains 422 chemical compounds, 24 different atoms, 66 atom types, and 4 types of bonds, containing on average 27 vertices and 28 edges per graph. All the experiments were run on a PC with Intel Core i5-2320, NVIDIA Tesla K20 (5GB DDR) and 8 GB RAM.

Fig. 3 shows the performance of *Algorithm* 2 with regard to different problem scales. It is clear that as the size of the graph increased, the accelerating effect became more and more significant. For graphs with 30 nodes, the average speedup was around 80 times. Table 1 compares the results of gSpan and the parallel gSpan with

[1] http://www.cs.ucsb.edu/~xyan/software/gSpan.htm

different support thresholds. It shows that the parallel gSpan can produce exactly identical results as the original gSpan. Table 2 presents the complete experiment results, which indicate that the parallel gSpan was more than one order of magnitude faster than the original gSpan. Note that the actual speedups also depend on the specific properties of the graphs.

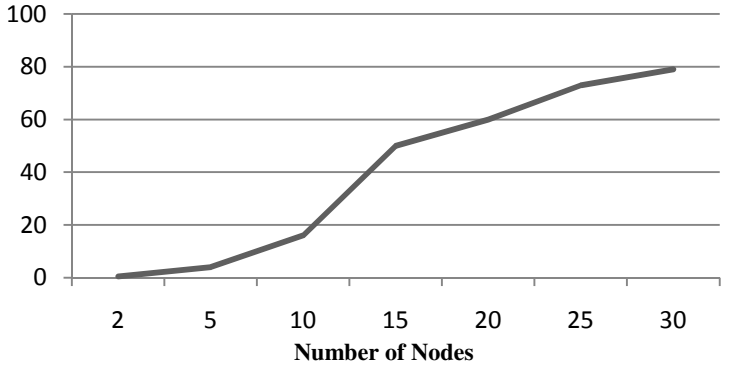


Fig. 3. Speedups on traversing a DFS code tree

Table 1. The accuracy of the parallel gSpan

Min_Sup	15%	18%	21%	23%	25%	35%
Accuracy	100%	100%	100%	100%	100%	100%

Table 2. The performance of the parallel gSpan

Min_Sup	15%	18%	21%	23%	25%	35%
CPU (s)	491.6	373.8	272.9	223.5	171.1	70.7
GPU (s)	32.4	23.6	17.7	15.2	11.8	6.8
Speedup	15.17	15.8	15.4	14.70	14.5	10.4

7 Conclusions

In this paper, we proposed a novel parallel method for traversing a DFS code tree and developed a parallel gSpan with the help of CUDA Dynamic Parallelism. With the careful implementation of the subgraph mining in gSpan, smart synchronization strategy and the optimized access of GPU memory, our method achieved up to 80 times speedup for traversing a DFS code tree and the parallel gSpan outperformed the original gSpan by an order of magnitude. As to future work, CUDA Dynamic Parallelism opens a new horizon for designing parallel algorithms on GPU and many graph mining methods such as FSG and FFSM [5] can potentially benefit from this advanced parallel computing architecture.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 60905030) and NVIDIA CUDA Teaching Center Program.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. ACM SIGMOD Record 22(2), 207–216 (1993)
2. Buehrer, G., Parthasarathy, S., Nguyen, A., et al.: Parallel Graph Mining on Shared Memory Architectures. Technical report, The Ohio State University (2005)
3. CUDA C Programming Guide, NVIDIA Corporation (2012), http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
4. Freeman, J.: Parallel Algorithms for Depth-First Search. Technical report (1991)
5. Huan, J., Wang, W., Prins, J.: Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism. In: 3rd IEEE International Conference on Data Mining, pp. 549–552 (2003)
6. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
7. Kepler GK110 Architecture Whitepaper, NVIDIA Corporation (2013), http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf
8. Kuramochi, M., Karypis, G.: Frequent Subgraph Discovery. In: 2001 IEEE International Conference on Data Mining, pp. 313–320 (2001)
9. Mannilla, H., Toivonen, H., Verkamo, I.: Discovering Frequent Episodes in Sequences. In: 1st International Conference on Knowledge Discovery and Data Mining, pp. 210–215 (1995)
10. Nijssen, S., Kok, J.N.: A Quickstart in Frequent Structure Mining Can Make a Difference. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 647–652 (2004)
11. Nijssen, S., Kok, J.N.: The Gaston Tool for Frequent Subgraph Mining. Electronic Notes in Theoretical Computer Science 127(1), 77–87 (2005)
12. Punin, J.R., Krishnamoorthy, M.S., Zaki, M.J.: LOGML: Log Markup Language for Web Usage Mining. In: Kohavi, R., Masand, B., Spiliopoulou, M., Srivastava, J. (eds.) WebKDD 2001. LNCS (LNAI), vol. 2356, pp. 88–112. Springer, Heidelberg (2002)
13. Raghavan, P.: Social Networks on the Web and in the Enterprise. In: Zhong, N., Yao, Y., Ohsuga, S., Liu, J. (eds.) WI 2001. LNCS (LNAI), vol. 2198, pp. 58–60. Springer, Heidelberg (2001)
14. Wang, C., Parthasarathy, S.: Parallel Algorithms for Mining Frequent Structural Motifs in Scientific Data. In: 18th ACM International Conference on Supercomputing, pp. 31–40 (2004)
15. Wittenbrink, C.M., Kilgariff, E., Prabhu, A.: Fermi GF100 GPU Architecture. IEEE Micro 31(2), 50–59 (2011)
16. Wörlein, M., Meinl, T., Fischer, I., Philippsen, M.: A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 392–403. Springer, Heidelberg (2005)
17. Yan, X., Han, J.: gSpan: Graph-Based Substructure Pattern Mining. In: 2002 International Conference on Data Mining, pp. 721–724 (2002)

Scaling Up Covariance Matrix Adaptation Evolution Strategy Using Cooperative Coevolution

Jinpeng Liu and Ke Tang*

USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China (USTC),
Hefei 230027, China
jpl@mail.ustc.edu.cn, ketang@ustc.edu.cn

Abstract. Covariance matrix adaptation evolution strategy (CMA-ES) has demonstrated competitive performance especially on multimodal non-separable problems. However, CMA-ES is not capable of dealing with problems having several hundreds dimensions. Motivated by that cooperative coevolution (CC) has scaled up many kinds of evolutionary algorithms (EAs) to high dimensional optimization problems effectively, we propose an algorithm called CC-CMA-ES which apply CC to CMA-ES in order to scale up CMA-ES to large scale problems. CC-CMA-ES adopts a new sampling scheme which does not divide population into small subpopulations and conducts mutation and crossover operations in subpopulation to generate offspring, but extracts a subspace Gaussian distribution from the global Gaussian distribution for subspace sampling. Also in CC-CMA-ES, two new decomposition strategies are proposed in order to balance exploration and exploitation. Lastly, an adaptive scheme is adopted to self-adapt appropriate decomposition strategy during evolution process. Experimental studies on a series of benchmark functions with different characteristic have been conducted and verified the excellent performance of our newly proposed algorithm and the effectiveness of the new decomposition strategies.

1 Introduction

Evolutionary algorithms (EAs) have been widely used in the field of numerical optimization [1]. However, they also suffer from "the curse of dimensionality" [2]. Cooperative coevolution (CC) [3] is an evolutionary framework based on divide-and-conquer strategy which has been used to scaling up some of EAs successfully [4–7]. DECC-G [4], proposed a decomposition scheme called random grouping which divides D-dimensional solution vector into m s-dimensional subvectors ($D = m \times s$) and conducts this many times each at the beginning

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 350–357, 2013.

of an optimization cycle to improves the probability of distributing interacting variables into a common subspace. MLCC [5] extended DECC-G by incorporating a self-adaptation strategy to select appropriate dimension for the subspace. CCPSO2 [6], imitating MLCC, adopted Cauchy and Gaussian based PSO as subspace optimizer. CCVIL [7] proposed a method to learn interacting variables other than random decomposition.

In this paper, we consider scaling up a representative kind of EAs, namely covariance matrix adaptation evolution strategy (CMA-ES) [8]. CMA-ES samples offspring via a multivariate Gaussian distribution and use offspring to update this distribution. It has demonstrated competitive performance compared with other EAs particular on multimodal non-separable problems [9]. However, CMA-ES also suffers from the curse of dimensionality. A variant of CMA-ES, namely sep-CMA-ES [10], has been proposed which can achieve linear time and space complexity. Unfortunately, experimental results have revealed the performance of sep-CMA-ES drops significantly with increasing dimensionality [11]. Motivated by all of those, we proposed an algorithm called CC-CMA-ES which apply CC to CMA-ES in order to scale up CMA-ES to large scale optimization problems. CC-CMA-ES adopts a new sampling scheme which does not divide population into small subpopulations and conducts mutation and crossover operations like that in DECC-G etc., but extracts a subspace Gaussian distribution from the global Gaussian distribution for subspace sampling. Also, two new decomposition strategies are proposed and an adaptive scheme is used to self-adapt appropriate decomposition strategy among two new decomposition strategies together with random based decomposition strategy during evolution process.

The rest of this paper is organized as follows. Section 2 describes CMA-ES briefly. Section 3 presents our new algorithm - CC-CMA-ES in detail, including new decomposition strategies and their adaptation scheme, sampling and update of distribution. Section 4 describes experimental setup and experimental results analysis. Finally, Section 5 concludes this paper.

2 Covariance Matrix Adaptation Evolution Strategy

As a respective of EAs, different from other EAs having mutation and crossover operations, CMA-ES firstly estimates a distribution from the samples and then takes sample to generate offspring using this distribution and repeats it until a stop criteria is satisfied.

$$x_k^{(g+1)} \sim N(x_w^g, (\sigma^g)^2 \cdot C^g), \qquad k = 1, 2, \cdots, \lambda \tag{1}$$

Equation 1 shows the sampling process at generation g. μ^g, C^g and σ^g are the Gaussian mean, covariance matrix and global step size respectively at generation g. CMA-ES has rotation invariant feature that is produced by C. This feature is the key of success for CMA-ES. When updating covariance matrix C, CMA-ES not only adopt current sample with better fitness but use history distribution information. Namely, CMA-ES combines *rank-*μ update and *rank-one* update [8]. The disadvantage of CMA-ES is its high time complexity, $O(D^3)$, where

D is the dimension of the solution vector, which avoids its application on high dimensional problems.

3 Proposed Algorithm

At the beginning of each optimization cycle of CC-CMA-ES, a decomposition strategy will be selected from a decomposition pool according their history performance and then the search space will be decomposed into several non-overlapping subspaces. Next, each subspace will be optimized sequently for a fixed number of fitness evaluations by a subspace CMA-ES whose parameters are extracted from a global CMA-ES and updated using the samples. At the end of each cycle, the performance record for the current selected decomposition strategy will be updated using its performance in this cycle. The same cycle will be conducted many times until a stop criteria is satisfied. The framework of CC-CMA-ES is showed in Algorithm 1 and each part of CC-CMA-ES will be described in detail as follows.

Algorithm 1. CC-CMA-ES

Require: space dimension dimension D, subspace dimension s, offspring size λ and fitness evaluation maximum $maxfes$

1: Initialize $D \times 1$ mean vector x_w, $D \times D$ covariance matrix C, global step size σ and other control parameters for global CMA-ES;
2: Initialize best solution *best* and the performance record;
3: Choose a decomposition strategy and decompose solution vector into m disjoint parts (so, each subspace has dimension $s = D/m$) (Sect. 3.2 in detail);
4: Set $sub = 1$ to start an optimization cycle;
5: Extract $s \times 1$ subspace mean vector x_{sub} and $s \times s$ covariance matrix C_{sub} from x_w and C and generate λ offspring using $N(x_{sub}, \sigma^2 \cdot C_{sub})$ (Sect. 3.3 in detail);
6: Evaluate offspring and update *best* if exists better offspring;
7: Update x_{sub}, C_{sub} and take them back to x_w, C (Sect. 3.4 in detail);
8: If the number of fitness evaluations reaches $maxfes$, stop. Else, if $sub < m$, $sub{+}{+}$ and go to step 5. Others, update performance record and go to step 3.

3.1 New Decomposition Strategies

In this paper, we propose two new decomposition strategies called Min-Variance decomposition strategy (MiVD) and Max-Variance decomposition strategy (MaVD) respectively. The emphasis of those new decomposition strategies is on the balance of exploration and exploitation.

MiVD and MaVD decompose space based on the diagonal of the covariance matrix. They firstly rank variables in the solution vector according to their variances in the diagonal of the covariance matrix and then separate the solution vector into several sub-vectors. MiVD decomposes variables with similar variances into a subspace to minimize the diversity among their variances. Opposite,

```
MaVD(D, m, C)
1 diagC ← diag(C);
2 s ← D/m;
3 subInfo ← ∅;
4 [sortedDiagC, sortedIndex]
  ← sort(diagC);
5 for i ← 1 to m
6    S_i ← sortedIndex(i : s : D);
7    subInfo ← subInfo ∪ {S_i};
8 end
9 return subInfo;
```

```
MiVD(D, m, C)
1 diagC ← diag(C);
2 s ← D/m;
3 subInfo ← ∅;
4 [sortedDiagC, sortedIndex]
  ← sort(diagC);
5 for i ← 1 to m
6    S_i ← sortedIndex((i − 1) · s + 1 : i · s);
7    subInfo ← subInfo ∪ {S_i};
8 end
9 return subInfo;
```

Fig. 1. Pseudocode of MaVD (left) and MiVD (right)

MaVD guarantees the diversity of the diagonal values of the variables in the same subspace as large as possible. Figure 1 shows their pseudocode. The sum of the volume of the tolerance region (hyperellipsoid) for each subspace with the same confidence by MiVD is larger than that of MaVD, which leads to MiVD is appropriate for exploration while MaVD for exploitation. From this perspective, random decomposition (RD) is a compromise between MiVD and MaVD.

3.2 Adaptive Decomposition Strategy Scheme

In CC-CMA-ES, three decomposition strategies – MaVD, MiVD and RD, will construct a pool and be used in an adaptive manner in order to accommodate various demands for decomposition strategy during environmental changeable evolution process like that in [5].

A performance record maintains the history performance for each decomposition strategy for a fixed number of optimization cycles (set to 5 in experiment), showed in Table 1. Rows indicate different decomposition strategies and columns indicate different cycles. Each item in the record demonstrates the fitness improvement rate of the best solution in this optimization cycle and is set to 1 initially. The probability of applying certain decomposition strategy is calculated by (3). At the beginning of each optimization cycle, we use stochastic universal selection method to select a decomposition strategy and record the fitness of the best solution found so far as $bestval_{old}$. After an optimization cycle, the fitness $bestval_{new}$ for new *best* will be obtained. The improvement rate for the decomposition strategy used in this cycle will be calculated according to (2) and the oldest history improvement of this decomposition strategy will be deleted to leave room for the new.

$$\delta_{ij} = \left| \frac{bestval_{new} - bestval_{old}}{bestval_{old}} \right| \tag{2}$$

$$probability_i = \frac{e^{\theta_i}}{\sum_{p=1}^{3} e^{\theta_p}} \qquad \text{where} \qquad \theta_i = \sum_{j=1}^{n} \delta_{ij} \ . \tag{3}$$

Table 1. Performance record

	$cycle_1$	$cycle_2$	$cycle_j$	$cycle_n$
Max-Variance decomposition	δ_{11}	δ_{12}	δ_{1j}	δ_{1n}
Random decomposition	δ_{21}	δ_{22}	δ_{2j}	δ_{2n}
Min-Variance decomposition	δ_{31}	δ_{32}	δ_{3j}	δ_{3n}

3.3 Subspace Sampling and Fitness Evaluation

After decomposition, subspaces have been obtained denoted as $S_i = \{x_{i1}, x_{i2}, \ldots, x_{is}\}$, $i = 1, 2, \ldots, m$, $D = m \times s$ where D is the dimension of the problem, m is the number of subspaces and s is the dimension of subspace. When optimizing subspace S_i, use Gaussian distribution $N(x_{S_i}, \sigma^2 \cdot C_{S_i})$ to generate λ offsprings. σ is the global step size. x_{S_i} and C_{S_i} are extracted from the global mean vector x_w and the covariance matrix C via the related dimension indexes in this subspace. For example, when $D = 4$, $s = 2$ and being optimized subspace S_i consists of dimension 1 and 3, namely $S_i = \{x_1, x_3\}$, the x_{S_i} and C_{S_i} will be extracted from the $D \times 1$ global mean vector x_w and the $D \times D$ covariance matrix C as follows.

$$\begin{bmatrix}1.8\\0.9\\3.2\\5.3\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}1.2 & 0.2 & 0.4 & 0.9\\0.2 & 1.7 & 0.7 & 0.85\\0.4 & 0.7 & 2.3 & 0.56\\0.9 & 0.85 & 0.56 & 2.8\end{bmatrix} \xrightarrow{extract} \begin{bmatrix}1.8\\3.2\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}1.2 & 0.4\\0.4 & 2.3\end{bmatrix}$$

We use greedy strategy [12] to evaluate the fitness for subspace samples. Namely, we take place of the values of the variables related to current subspace in the best solution *best* found so far using sampled offspring and use the fitness of this complete solution as the fitness of offspring. After optimization of each subspace, the best solution *best* will be updated if better solution is found.

3.4 Updating CMA-ES Parameters

x_{S_i}, C_{S_i} and σ will be updated using offspring [8]. Then we will take place the portion of the global mean x_w and covariance matrix C related to current subspace while other portions are fixed by those subspace parameters to guarantee the newest correlation information for subsequent usage, that is, this procedure is the inverse process of extracting x_{S_i} and C_{S_i} from x_w and C. As the example in 3.3, after optimizing subspace $S_i = 1, 3$, updating the subspace mean vector x_{S_i} and the subspace covariance matrix C_{S_i} and take those back to the global mean vector x_w and the global covariance matrix C is showed as follows.

$$\begin{bmatrix}1.8\\3.2\end{bmatrix} \quad \begin{bmatrix}1.2 & 0.4\\0.4 & 2.3\end{bmatrix} \xrightarrow{update} \begin{bmatrix}2.8\\4.2\end{bmatrix} \quad \begin{bmatrix}2.2 & 0.88\\0.88 & 1.9\end{bmatrix} \xrightarrow{reverse} \begin{bmatrix}2.8\\0.9\\4.2\\5.3\end{bmatrix} \quad \begin{bmatrix}2.2 & 0.2 & 0.88 & 0.9\\0.2 & 1.7 & 0.7 & 0.85\\0.88 & 0.7 & 4.2 & 0.56\\0.9 & 0.85 & 0.56 & 2.8\end{bmatrix}$$

Table 2. The results are the mean best values averaged over 25 independent runs and standard deviation. Wilcoxon signed rank test with significance 0.05 for CC-CMA-ES and each other algorithm have been conducted and statistically significant results are labeled by superscript *.

Function	Dim		DECC-G	MLCC	CCPSO2	CC-CMA-ES
f_1	1000	mean	$6.28e-06^*$	**1.24e-23***	$2.94e-05^*$	$5.77e-09$
		std	$9.62e-06$	$4.94e-23$	$4.39e-05$	$1.00e-09$
f_2	1000	mean	$1.32e+03$	**5.67e+00***	$6.63e+00^*$	$1.33e+03$
		std	$5.23e+01$	$8.92e+00$	$3.14e+01$	$1.11e+02$
f_3	1000	mean	$1.11e+00^*$	$1.41e-10$	$8.41e-06^*$	**1.51e-13**
		std	$2.70e-01$	$7.02e-10$	$7.54e-06$	$6.73e-15$
f_4	1000	mean	$2.25e+11^*$	$1.03e+11^*$	$7.21e+11^*$	**2.20e+09**
		std	$1.43e+11$	$4.41e+10$	$9.66e+11$	$1.31e+09$
f_5	1000	mean	$7.28e+14$	$7.28e+14$	$7.28e+14^*$	$7.28e+14$
		std	$1.65e+07$	$1.68e+06$	$3.17e+05$	$4.65e+06$
f_6	1000	mean	**1.84e+05***	$9.07e+05$	$8.60e+05$	$5.83e+05$
		std	$2.89e+04$	$1.84e+05$	$3.03e+05$	$4.79e+05$
f_7	1000	mean	$9.92e+08^*$	$8.64e+08^*$	$3.21e+09^*$	**7.45e+06**
		std	$5.62e+08$	$8.10e+08$	$2.18e+09$	$1.21e+07$
f_8	1000	mean	$6.82e+15^*$	$5.25e+15^*$	$1.47e+16^*$	**3.88e+14**
		std	$3.17e+15$	$2.83e+15$	$1.47e+16$	$2.87e+14$
f_9	1000	mean	$5.78e+08^*$	$8.51e+08^*$	$7.66e+08^*$	**3.71e+08**
		std	$1.52e+08$	$1.73e+08$	$3.82e+08$	$1.83e+08$
f_{10}	1000	mean	$2.91e+07^*$	$5.08e+07^*$	$4.02e+07^*$	**7.55e+05**
		std	$1.14e+07$	$2.51e+07$	$3.82e+07$	$5.02e+05$
f_{11}	1000	mean	$1.21e+11^*$	$1.08e+11^*$	$4.49e+11^*$	**1.59e+08**
		std	$7.35e+10$	$8.53e+10$	$3.04e+11$	$1.47e+08$
f_{12}	1000	mean	$9.58e+03^*$	$3.33e+03^*$	$1.46e+03$	**1.27e+03**
		std	$6.30e+03$	$2.98e+03$	$5.17e+02$	$4.26e+02$
f_{13}	905	mean	$8.88e+09^*$	$7.20e+09^*$	$2.73e+10^*$	**6.70e+08**
		std	$3.21e+09$	$2.96e+09$	$7.43e+09$	$1.14e+09$
f_{14}	905	mean	$1.34e+11^*$	$1.09e+11^*$	$5.50e+11^*$	**7.10e+07**
		std	$5.18e+10$	$6.98e+10$	$2.55e+11$	$1.25e+08$
f_{15}	1000	mean	$1.20e+07^*$	**7.44e+06***	$5.30e+08^*$	$3.03e+07$
		std	$9.91e+05$	$1.33e+06$	$9.26e+08$	$6.08e+06$

4 Experimental Studies

4.1 Experimental Setup

We have chosen the benchmark functions provided by *CEC2013 Special Session on large scale global optimization* [13] for evaluating CC-CMA-ES and comparisons with other algorithms. The algorithms for comparisons consist of DECC-G, MLCC and CCPSO2 all of which are state-of-art CC based algorithms and have demonstrated remarkable performance on large scale global optimization. The control parameters used in those algorithms could be found in [4–6]. For CC-CMA-ES, the size of offspring is set to 50 and the number of subspaces is set

to 20. We set the maximum fitness evaluations for each iteration of optimizing a certain subspace to $10 \times s$ where s is the dimension of this subspace. For fair comparison, the stop criterion for all algorithms is the same number of fitness evaluations $3e6$ and each algorithm will run for 25 times independently.

4.2 Experimental Results

From Table 2, CC-CMA-ES performs significantly better on eleven out of fifteen functions compared with DECC-G, which demonstrates that CC-CMA-ES is more effective in capturing the correlation between variables through the use of covariance matrix. CC-CMA-ES was outperformed significantly by DECC-G on function 6 and 15 where function 6 are partially separable function and function 15 are fully non-separable function. The reason may be the decomposition of space in CC-CMA-ES results into the correlation among variables could not be updated timely and accurately. CC-CMA-ES outperforms MLCC significantly on nine functions and was outperformed significantly on function 1, 2 and 15. It is noteworthy that function 1 and 2 are fully-separable functions. And on the other fully-separable function 3, CC-CMA-ES does not outperform MLCC significantly. Combing the results of DECC-G, we could find the effectiveness of MLCC derived from the self-adaptative scheme of subspace dimension. The comparison result on function 15 is similar to that of DECC-G, which reveals the disadvantage of CC-CMA-ES on fully non-separable problems. CC-CMA-ES outperforms CCPSO2 significantly on all functions except function 2 and 6. The phenomenon of CC-CMA-ES was outperformed on function 2 shows similar result to that of MLCC, which demonstrated the relative weakness of CC-CMA-ES on fully separable functions compared with its strength on partially additive separable functions.

It is noteworthy CC-CMA-ES outperforms best on all three overlapping functions 12, 13 and 14. Overlapping functions were included in *CEC2013* benchmark functions for the first time, and no related results on those functions have been reported. The effectiveness of CC-CMA-ES on overlapping functions shows the advantage of combing CMA-ES and CC framework where CMA-ES can capture the interdependence among variables and CC can alleviate the hardness of CMA-ES on high dimension problems.

5 Conclusion

In this paper, we propose a new algorithm CC-CMA-ES which scales up CMA-ES to high dimensional problems using CC. In this algorithm, two new decomposition strategies are proposed which are based on the diagonal of the covariance matrix of CMA-ES to keep the balance between exploration and exploitation in evolution process. To coordinate different decomposition strategy, an adaptive decomposition strategy scheme is adopted which could select appropriate decomposition strategy adaptively. Comprehensive experiment studies on large scale problems demonstrated the effectiveness of our proposed algorithm and

verified the validity of the new decomposition strategies and adaptive decomposition strategy scheme. Although the effectiveness of the two new decomposition strategies has been verified on the whole, it still needs studies on fully non-separable problems. Moreover, those two new decomposition strategies could be used in other EAs in order to study their performance in various circumstances.

Acknowledgment. This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant 61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16.

References

1. Sarker, R., Mohammadian, M., Yao, X.: Evolutionary optimization, vol. 48. Kluwer Academic Pub. (2002)
2. Van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 225–239 (2004)
3. Potter, M.A., De Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
4. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Sciences 178(15), 2985–2999 (2008)
5. Yang, Z., Tang, K., Yao, X.: Multilevel cooperative coevolution for large scale optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1663–1670. IEEE (2008)
6. Li, X., Yao, Y.: Cooperatively coevolving particle swarms for large scale optimization. IEEE Transactions on Evolutionary Computation 16(2), 1–15 (2011)
7. Chen, W., Weise, T., Yang, Z., Tang, K.: Large-scale global optimization using cooperative coevolution with variable interaction learning. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 300–309. Springer, Heidelberg (2010)
8. Hansen, N.: The cma evolution strategy: A tutorial. Vu le 29 (2005)
9. Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 282–291. Springer, Heidelberg (2004)
10. Ros, R., Hansen, N.: A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 296–305. Springer, Heidelberg (2008)
11. Omidvar, M.N., Li, X.: A comparative study of cma-es on large scale global optimisation. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 303–312. Springer, Heidelberg (2010)
12. Wiegand, R.P., Liles, W.C., De Jong, K.A.: An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), vol. 2611, pp. 1235–1245 (2001)
13. Li, X., Tang, K., Omidvar, M.N., Yang, Z., Qin, K., China, H.: Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Gene 7, 33 (2013)

Gradient Boosting-Based Negative Correlation Learning

Lunjun Wan, Ke Tang*, and Rui Wang

USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China (USTC),
Hefei 230027, China
wljcai@mail.ustc.edu.cn, ketang@ustc.edu.cn, wrui1108@mail.ustc.edu.cn

Abstract. Negative correlation learning (NCL) is a useful ensemble learning approach, and has been used for neural network ensembles. In this paper, a new NCL algorithm, NCL.GBM is proposed, which uses gradient boosting machine (GBM) as the base learner. First, the feasibility of combining NCL and GBM is analysed. Then, we describe in detail how to apply negative correlation learning onto GMB. Empirical results show that NCL.GBM does have the ability to produce models with lower correlation and can usually result in lower generalization error than the original GBM.

Keywords: Negative correlation learning, gradient boosting machine.

1 Introduction

The simple idea of ensemble learning is combining a batch of "weak" learning models to get a "strong" model [13]. Several theories have been published to explain the reasons account for the success of ensemble methods [14] [8] [15]. A common understanding of these theories is that the diversity among base classifiers is key to the success of an ensemble model.

To train an ensemble with diverse classifiers explicitly, Liu and Yao firstly proposed a general ensemble learning framework, named negative correlation learning (NCL) in [9], which aims to train ensemble models by keeping the correlation among base classifiers as smaller as possible. NCL poses a correlation penalty term onto the loss function (i.e., the mean square error) of standard neural networks, which will be lager when the output of the neural network is correlated with that of other neural networks in the ensemble. Therefore, diverse classifiers would be trained explicitly given the loss of each neural networks is minimized. Adding a correlation penalty term into loss function is not the only way to bring negative correlation strategy into ensemble learning. Negative Correlation Learning via Correlation-Corrected Data (NCCD) [3] which also uses

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 358–365, 2013.

neural networks as base learners, introduces diversity penalty into the training data. That is, NCCD constructs a sets of negative-corrected training data. AdaBoost.NC [16] introduces error correlation information into the weights of training data when building each base classifier of AdaBoost [5]. All these works utilize the idea of negative correlation to maintain diversity of ensemble.

However, the original NCL is only implemented with neural networks as the base learner, since it is easy to add a correlation penalty term to the original loss function of neural networks, and the result new loss function can also be minimized with standard neural networks' training algorithm. Therefore, learning models which are not trained with an explicit loss function, such as tree-based methods, are not suitable to NCL directly. Obviously this is a big obstacle blocking NCL to be a more adaptive ensemble learning method. Besides tree-based method shows an advantage over neural network because it is more interpretable. So in this paper, we aim to extend the general NCL framework to tree-based type classification algorithms.

We propose a new negative correlation learning algorithm based on gradient boosting machine (NCL.GBM) which replaces the base learners from neural networks to gradient boosting machine (GBM) [7]. GBM fits an additive model in a forward stage-wise manner using gradient descent method. The additive model could be an ensemble of regression trees. We will explain why GBMs can be used in NCL and how to train them using NCL algorithm framework. We test our approach on some classification tasks. The experiment results show that NCL.GBM owns significant higher performance compared to the original GBM algorithm in most cases.

The rest of our paper is organized as follows: In Section 2, we take a NCL algorithm as an example to illustrate what negative correlation learning is and how it works. In Section 3, GBM will be introduced firstly. Then we present our approach in detail about why we choose GBM as the base learner, and how to train them. In Section 4, we do experiments to analyse the generalization performance of NCL.GBM comparing to GBM. We conclude our work in Section 5 with a summary and future works.

2 Overview of Negative Correlation Learning

In this section, we introduce a successful NCL algorithms: Cooperative Ensemble Learning System (CELS) [10] to explain how NCL works. CELS is designed for neural network ensemble. Suppose there is a training set

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$$

where $\mathbf{x}_i \in R^d$, $y_i \in \{+1, -1\}$, N is the size of training set. The output of the CLES on sample $\mathbf{x}_i$ is

$$F(\mathbf{x}_i) = \frac{1}{M} \sum_{m=1}^{M} f_m(\mathbf{x}_i) \tag{1}$$

where M is the number of neural network estimators, and $f_m(\mathbf{x}_i)$ is the output of the m-th neural network on sample $\mathbf{x}_i$. By introducing a correlation penalty term in the loss function, CELS can create negatively correlated neural networks. The loss function for the m-th neural network in CELS is

$$E_m = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(f_m(\mathbf{x}_i) - y_i)^2 + \frac{1}{N}\sum_{i=1}^{N}\lambda p_m(\mathbf{x}_i) \tag{2}$$

where p_m is a correlation penalty function, parameter $0 \leq \lambda \leq 1$ is used to control the strength of the penalty. The first term of Eq. (2) is the empirical loss function derived from the standard neural network algorithm. The penalty term p_m goes like

$$p_m(\mathbf{x}_i) = (f_m(\mathbf{x}_i) - F(\mathbf{x}_i))\sum_{j\neq m}(f_j(\mathbf{x}_i) - F(\mathbf{x}_i)). \tag{3}$$

Hence, minimizing p_m is to encourage different individual networks to learn different aspects of the training data set so that the ensemble can learn the whole training data set better. All the individual neural networks of CELS are trained synchronously and interactively and share the same training process except for using different loss functions.

3 Our Approach

3.1 Friedman's Gradient Boosting Machine

GBM is a boosting algorithms using the connection between boosting and optimization [6][2][11]. Friedman et al. [6] have proved that the most commonly used boosting algorithm: AdaBoost [4] is actually fitting an additive model in a forward stage-wise manner and can be regarded as an optimization algorithm minimizing a specific exponential loss function. GBM generalizes boosting by replacing the exponential loss function to arbitrary differentiable loss function, such as least squares, least absolute deviation and logistic binomial log-likelihood. Each weak classifier of boosting is a single term of the additive model. Considering the ensemble of weak classifiers as a single classifier, GBM searches for the best classifier in function space to minimize the loss function using gradient descent method. The framework of gradient boosting machine is shown in Algorithm 1, where L is the loss function, F is the output of GBM and ν is the learning rate parameter.

3.2 Gradient Boosting-Based NCL Algorithm

We develop a novel gradient boosting-based negative correlation learning algorithm called NCL.GBM. NCL.GBM aims to create an ensemble of GBMs. Just like NCL for neural network, NCL.GBM introduces the correlation penalty term

Algorithm 1 Gradient Boosting algorithm

1: $\widehat{F}(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho)$
2: **for** t in 1,2,...,T **do**
3: $z_i = -\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}|_{F(\mathbf{x}_i)=\widehat{F}(\mathbf{x}_i)}, i = 1, ..., N$
4: $\mathbf{a} = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [z_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
5: $\rho = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \widehat{F}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}))$
6: $\widehat{F}(\mathbf{x}) = \widehat{F}(\mathbf{x}) + \nu * \rho h(\mathbf{x})$
7: **end for**

into the loss function of GBM. Here we take the squared-error loss, then the loss function of the m-th GBM goes like

$$\Phi_m(y, f_m(\mathbf{x})) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} (f_m(\mathbf{x}_i) - y_i)^2 + \frac{1}{N} \sum_{i=1}^{N} \lambda p_m(\mathbf{x}_i) \tag{4}$$

where $p_m(\mathbf{x}_i)$ is the same as Eq. (3).

Algorithm 2 presents the pseudo-code of NCL.GBM. The first step of training progress is initializing each individual GBMs with diverse values. Unlike neural network, GBM is nearly a deterministic algorithm when both training data set and parameters are fixed. So, in order to generate diversiform GBMs, diversity should be ensured initially. A simple way to solve this problem is to initialize the output of each GBM with a random value. After that, we train these GBMs synchronously and interactively for T epochs by gradient descent method to minimize the loss function. In each epoch, the negative gradient of each training sample is calculated, then a regression tree [12] with K terminal nodes is built to fit these gradient values. The predictions of terminal nodes are recomputed to minimize the loss function on the training data. The negative gradient in step 5 goes like

$$z_i = y_i - \tilde{f}_m(\mathbf{x}_i) + \lambda(\tilde{f}_m(\mathbf{x}_i) - F(\mathbf{x}_i)) \tag{5}$$

and the terminal node prediction in step 7 goes like

$$\rho_k = \frac{1}{|S_k|} \sum_{\mathbf{x}_i \in S_k} (y_i - \tilde{f}_m(\mathbf{x}_i) + \lambda(\tilde{f}_m(\mathbf{x}_i) - F(\mathbf{x}_i))) \tag{6}$$

4 Experimental Studies

4.1 NCL.GBM vs GBM

This section presents the experiments between NCL.GBM and GBM. This work is a preliminary work, so we focus on binary classification problem. 10 data sets chosen from UCI machine learning repository [1] are used. The detailed

Algorithm 2 Gradient boosting-based NCL algorithm

1: Randomly initialize $\widehat{f}_m(\mathbf{x}), m = 1, ..., M$
2: $F(\mathbf{x}_i) = \frac{1}{M}\sum_{m=1}^{M}\widehat{f}_m(\mathbf{x}_i), i = 1, ..., N$
3: **for** t in 1,2,...,T **do**
4: **for** m in 1,2,...,M **do**
5: Compute negative gradient

$$z_i = -\frac{\partial \Phi_m(y_i, f_m(\mathbf{x}_i))}{\partial f_m(\mathbf{x}_i)}|_{f_m(\mathbf{x}_i)=\widehat{f}_m(\mathbf{x}_i)}, i = 1, ..., N$$

6: Fit a regression tree with K terminal nodes, predicting z_i from the co-variate $\mathbf{x}_i$
7: Compute optimal terminal node prediction as

$$\rho_k = \arg\min_{\rho_k} \sum_{\mathbf{x}_i \in S_k} \Phi_m(y_i, \widehat{f}_m(\mathbf{x}_i) + \rho_k)$$

where S_k is the set of $\mathbf{x}$s that drop in terminal node k, $k = 1, ..., K$.
8: $\widehat{f}_m(\mathbf{x}_i) = \widehat{f}_m(\mathbf{x}_i) + \nu * \rho_k, \mathbf{x}_i \in S_k, k = 1, ..., K$.
9: **end for**
10: $F(\mathbf{x}_i) = \frac{1}{M}\sum_{m=1}^{M}\widehat{f}_m(\mathbf{x}_i), i = 1, ..., N$
11: **end for**

information about the data sets is presented in Table 1. Parameters that used to train the algorithms are as follows,

- GBM: we set K as 3, the maximum of T as 500 and ν to $\{0.05, 0.1, 0.5, 1\}$.
- NCL.GBM: we set M to $\{3, 5, 7, 9, 11\}$ and λ to $\{0.1, 0.25, 0.5, 0.75, 1.0\}$.

Table 1. Summary of classification data sets

Data Sets	Feature Numbers	Sample Numbers
australian	14	307:383
bcw	9	458:241
heart	13	150:120
ionosphere	34	225:126
magic04	10	12332:6688
mammographic	5	445:516
promoters	57	53:53
sonar	60	97:111
spambase	57	1813:2788
wpbc	33	47:151

Two-thirds of data samples are used to adjust the parameters. We run GBM using 5 folds cross-validation, T and ν that get the highest average validation accuracy are kept for GBM. To be fair, we set the same value of T and ν to NCL.GBM. Using the same methodology, we get M and λ for NCL.GBM. With

these chosen parameters we conduct 10 times of 10-fold cross-validation method to evaluate GBM and NCL.GBM. We record the mean classification error rates over 100 runs. Wilcoxon signed-rank test at 0.05 significant level is employed to check whether the performance of NCL.GBM is significant different from that of GBM.

The experiment results are listed in Table 2. We can see that NCL.GBM shows statistically lower error rates than GBM on most of the data sets, which suggests that GBM can be used in NCL and negative correlation method has the ability to improve the generalization performance of GBM.

Table 2. Comparison of NCL.GBM and GBM on the two-class data sets by mean(standard deviation) error percent of 100 runs. A Win-Loss-Tie summarization based on mean error rate and Wilcoxon signed-rank sum test (0.05 significance level, significant difference is marked by '*') is attached at the bottom of the table.

data	GBM	NCL.GBM
australian	14.65±3.83	14.46±3.84*
bcw	3.99±2.46	3.89±2.51
heart	17.88±7.51	16.24±7.72*
ionosphere	7.41±4.75	7.12±4.58*
magic04	14.91±0.81	14.54±0.75*
mammographic	16.28±3.55	16.34±3.74
promoters	11.27±9.20	10.76±8.79*
sonar	19.27±8.23	16.4584±7.92*
spambase	6.37±1.20	5.70±1.11*
wdbc	7.42±3.23	5.53±3.01*
Significant W-L-T	-	0/7/2

4.2 Correlation and Performance Analysis

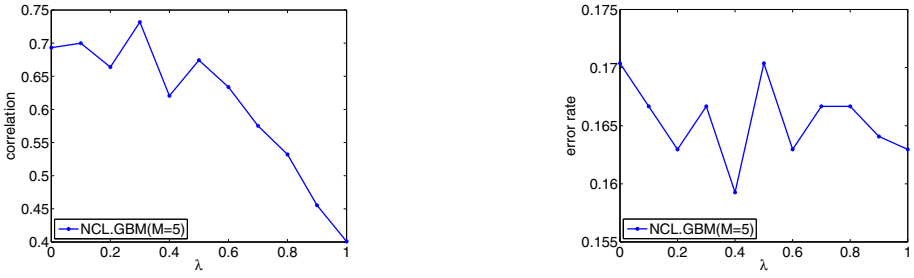
In order to observe the impact of correlation penalty term, we use person correlation coefficient to evaluate the correlations among individual GBMs of NCL.GBM. The correlation between the i-th GBM and the j-th GBM is given by

$$Cor_{ij} = \frac{\sum_{n=1}^{N}(f_i(n) - \overline{f_i})(f_j(n) - \overline{f_j})}{\sqrt{\sum_{n=1}^{N}(f_i(n) - \overline{f_i})^2}\sqrt{\sum_{n=1}^{N}(f_j(n) - \overline{f_j})^2}} \tag{7}$$

where $f_i(n)$ is the output of the i-th GBM on the nth sample in the testing set, $\overline{f_i}$ is the average output of i-th GBM over the whole testing samples. N is the size of testing set.

Figure 1(a) shows the average correlation value among GBMs of NCL.GBM with different λ on data set Heart. Here the ensemble model contains five GBMs. We can see that although there are some ups and downs between $\lambda = 0.2$ and $\lambda = 0.5$, it is still obviously that the average correlation is reducing when λ

increases from 0.0 to 1.0. It suggests that the correlation penalty term is working in NCL.GBM. These ups and downs may be caused by the randomness of experiment.



(a) average correlation (b) classification error

Fig. 1. Pearson correlation and classification error of NCL.GBM with different λ values on data set Heart

We also record the classification error rate of NCL.GBM when λ increases from 0.0 to 1.0 as showing in Figure 1(b). Although lambda shows limit effect here, we can still see that, the classification error can be reduced when λ is set to appropriate value (e.g., 0.4 in this case). Combining the two pictures together, we conclude that there is a trade-off between diversity and accuracy. In other words, to achieve best generalization performance, λ should be neither too big nor too small.

5 Conclusions

In this paper, we apply NCL to Friedman's gradient boosting machine and propose a new NCL algorithm, NCL.GBM. We conduct some empirical experiments to compare our approach to GBM. The results show that our approach can usually result in better classification accuracy on most cases than GBM. The correlations among the individual GBMs of NCL.GBM are calculated and explain that negative correlation learning really works in NCL.GBM.

However there is still some issues that need to be solved. First, both NCL.GBM and GBM are ensemble methods, but in our experiments NCL.GBM builds more regression trees than GBM does. We wonder if we use an ensemble pruning method to reduce the size of NCL.GBM, whether the generalization performance of NCL.GBM could be kept or not. If it works, then we can get a new ensemble which has the same size to GBM but shows superior performance. Second, this paper is a preliminary work, we haven't compared NCL.GBM with other NCL algorithms. So our future works will be finding a suitable ensemble pruning method so that both the efficiency and performance of NCL.GBM can be improved and doing more comparison experiments with some other NCL algorithms to test our approach.

Acknowledgement. This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant 61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16.

References

1. Bache, K., Lichman, M.: UCI machine learning repository (2013)
2. Breiman, L.: Prediction games and arcing algorithms. Neural Computation 11(7), 1493–1517 (1999)
3. Chan, Z.S.H., Kasabov, N.: A preliminary study on negative correlation learning via correlation-corrected data (nccd). Neural Processing Letters 21(3), 207–214 (2005)
4. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. In: Machine Learning-International Workshop then Conference, pp. 148–156. Morgan Kaufmann Publishers, Inc. (1996)
5. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
6. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The Annals of Statistics 28(2), 337–407 (2000)
7. Friedman, J.H.: Greedy function approximation: a gradient boosting machine (english summary). Ann. Statist. 29(5), 1189–1232 (2001)
8. Lam, L.: Classifier combinations: Implementations and theoretical issues. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 77–86. Springer, Heidelberg (2000)
9. Liu, Y., Yao, X.: Ensemble learning via negative correlation. Neural Networks 12(10), 1399–1404 (1999)
10. Liu, Y., Yao, X.: Simultaneous training of negatively correlated neural networks in an ensemble. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29(6), 716–725 (1999)
11. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: Advances in Neural Information Processing Systems 12, pp. 512–518. MIT Press (2000)
12. Breiman, L., Friedman, J.H., Olshen, R.A., StoneC.J.: Classification and regression trees. Wadsworth International Group (1984)
13. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
14. Bruce, E.R.: Ensemble learning using decorrelated neural networks. Connection Science 8(3-4), 373–384 (1996)
15. Tang, E.K., Suganthan, P.N., Yao, X.: An analysis of diversity measures. Machine Learning 65(1), 247–271 (2006)
16. Wang, S., Chen, H., Yao, X.: Negative correlation learning for classification ensembles. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010)

Metamodel Assisted Mixed-Integer Evolution Strategies Based on Kendall Rank Correlation Coefficient

Lili Zhuang[1], Ke Tang[1,⋆], and Yaochu Jin[2]

[1] USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China (USTC),
Hefei 230027, China
zhuangl@mail.ustc.edu.cn, ketang@ustc.edu.cn
[2] Department of Computing, University of Surrey,
Guildford, Surrey GU2 7XH, United Kingdom
yaochu.jin@surrey.ac.uk

Abstract. Although mixed-integer evolution strategies (MIES) have been successfully applied to optimization of mixed-integer problems, they may encounter challenges when fitness evaluations are time consuming. In this paper, we propose to use a radial-basis-function network (RBFN) trained based on the rank correlation coefficient distance metric to assist MIES. For the distance metric of the RBFN, we modified a heterogeneous metric (HEOM) by multiplying the weight for each dimension. Whilst the standard RBFN aims to approximate the fitness accurately, the proposed RBFN tries to rank the individuals (according to their fitness) correctly. Kendall rank correlation Coefficient (RCC) is adopted to measure the degree of rank correlation between the fitness and each variable. The higher the rank similarity with fitness, the greater the weight one variable will be given. Experimental results show the efficacy of the MIES assisted by the RBFN trained by maximizing the RCC performs.

1 Introduction

Evolution Strategies (ES) [1] are a branch of Evolutionary Algorithms (EA) [2]. Successful as they are, ESs have encountered challenges, one of which is the heterogeneity of the decision variables. There are some real-world optimization problems, whose decision variables are of different types. It usually contains continuous variables, integer variables and nominal discrete variables simultaneously. Canonical Evolution Strategies usually work on optimization problems of homogeneous (typically continuous) decision variables only.

Mixed-integer evolution strategies (MIES) [3] were proposed by Emmerich *et al* to optimize the rigorous process simulators which is a mixed-integer optimization problem in chemical plant design. MIES can deal with different variable

⋆ Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 366–375, 2013.

types simultaneously, which usually include continuous, integer and nominal discrete. In [3–5], MIES have been employed to successfully solve mixed-integer optimization problems occurring in optical filter design, rigorous process simulators for chemical plant design and image analysis agent for intravascular ultrasound image analysis.

Another challenge to Evolution Strategies is the difficulty in fitness evaluations [6]. As discussed in [6], in many real-world problems, fitness evaluations need a high complexity of performance analyses, which means each single fitness evaluation is highly time-consuming. In some other cases, explicit fitness functions do not exist, therefore human experts are needed for assigning a fitness value to a candidate solution. Both huge time consumption for fitness evaluations and fatigue of human experts will prevent Evolution Strategies from being applied to a wider range of problems. Using a metamodel (also known as surrogates) to estimate the fitness values is a common approach to addressing this kind of problems [6]. Typical metamodels include polynomial models, kriging models, neural networks and support vector machines [7].

The difficulty in fitness evaluations in real-world applications is also a challenge that MIES faces. A straightforward idea for addressing this problem is to use a metamodel-assisted MIES. However, most research on metamodel assisted Evolution Strategies focus on continuous optimization problems [7]. To the best of our knowledge, only one paper has been reported on developing metamodels for MIES [8]. In [8], Li *et al* chose radial-basis-function networks (RBFNs) [9] as the metamodel, and modified the canonical RBFN to make it more suited for mixed-integer search spaces by introducing a heterogeneous distance metric.

In this paper, we propose a new RBFN to assist MIES. The distance metric of the proposed RBFN is based on the Kendall rank correlation coefficient (RCC) [10]. Before obtaining the distance between two individuals, we first determine the weights for each variable according to the Kendall RCC between variables and the true fitness values. Since the distance metric is related to the rank, we hope that the new RBFN will help MIES to select better individuals, thereby improving the performance of the RBFN assisted MIES.

This paper is organized as follows. In Section 2.1, we introduce MIES together with a formal statment of Mixed-integer optimization problems. Recombination and mutation operators for MIES proposed by Li in [3] are discussed. RBFN assisted MIES are described in Section 2.2, where the modified the distance metric for RBFN proposed in [8] is also presented. In Section 3, we introduce the Kendall rank correlation coefficient [10] and propose a RCC based RBFN for assisting MIES. The performance of the new algorithm is verified on four test problems in Section 4. Section 5 concludes this paper.

2 RBFN Assisted MIES

2.1 Mixed-Integer Evolution Strategies

MIES can deal with different types of variables simultaneously, which usually include continuous, integer and nominal discrete. A mixed-integer global opti-

mization problem can be defined as follows [8]:

$$f(r_1, ..., r_{n_r}, z_1, ..., z_{n_z}, d_1, ..., d_{n_d}) \rightarrow \min \tag{1}$$

$$\text{subject to:}$$
$$r_i \in [r_i^{min}, r_i^{max}] \subset \mathbb{R}, i = 1, ..., n_r$$
$$z_i \in [z_i^{min}, z_i^{max}] \subset \mathbb{Z}, i = 1, ..., n_z$$
$$d_i \in D_i = \{d_{i,1}, ..., d_{i,|D_i|}\}, i = 1, ..., n_d$$

Here, r denotes the continuous variables, z integer variables, and d the nominal discrete variables. r_i denotes the ith continuous variable, z_i the ith integer variable, and d_i the ith nominal discrete variable. n_r, n_z and n_d is the number of continuous, integer, and nominal discrete variables, respectively. D_i denotes a set of nominal discrete values. The fitness function is denoted by f.

An individual in Evolution Strategies is denoted as [3]:

$$\boldsymbol{a} = (r_1, ..., r_{n_r}, z_1, ..., z_{n_z}, d_1, ..., d_{n_d}, \sigma_1, ..., \sigma_{n_\sigma}, \varsigma_1, ..., \varsigma_{n_\varsigma}, \rho_1, ..., \rho_{n_\rho})$$

The parameters $r_1, ..., r_{n_r}$, $z_1, ..., z_{n_z}$, $d_1, ..., d_{n_d}$ are called object parameters, correspond to the variables of mixed-integer optimization, while $\sigma_1, ..., \sigma_{n_\sigma}$, $\varsigma_1, ..., \varsigma_{n_\varsigma}$, $\rho_1, ..., \rho_{n_\rho}$ are strategy parameters for Evolution Strategies. $\sigma_1, ..., \sigma_{n_\sigma}$ are the average step size for continuous values, $\varsigma_1, ..., \varsigma_{n_\varsigma}$ are step size for integer values and $\rho_1, ..., \rho_{n_\rho}$ are the mutation probabilities for nominal discrete values.

There are two most widely used classes of recombination used in ES: discrete recombination, sometimes also referred to as dominant recombination, and intermediate recombination [1]. In this paper, we adopted dominant recombination for object parameters and intermediate recombination for strategy parameters. For each object parameter of offspring individual, dominant recombination chooses the object parameter from parents with a equal probability. By contrast, for each strategy parameter of the offspring individual, intermediate recombination obtains the mean of the strategy parameter from all recombination parents.

Different variable types need different mutation operators. To make the mutation operator suited for mixed-integer optimization problems, Emmerich *et al.* proposed a new mutation operator in [3]. This mutation operator is combined with the standard mutations for continuous, integer and nominal discrete, as described in [11–13].

Algorithm 1 presents the detail of the mutation, where τ_g denotes the global learning rate and τ_l the local learning rate. The recommended settings [3] are $\tau_l = 1/\sqrt{2\sqrt{n_r}}$ and $\tau_g = 1/\sqrt{2n_r}$. $U(0,1)$ denotes uniform distribution and $N(0,1)$ denotes the standardized normal distribution. $T^z_{[a,b]}$ is a transformation function for integer parameters [3]. Transformation makes sure that the values are within the boundaries. The details of the transformation are shown in [3].

2.2 RBFN Assisted MIES

MIES has been applied successfully to real-world problems [3,5]. However, MIES has also encountered some challenges such as the time consuming of fitness

/*Mutation of continuous values*/
for $i = 1, ..., n_r$ **do**
  $\sigma_i' \leftarrow \sigma_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $r_i' \leftarrow r_i + N(0, s_i')$
end
/*Mutation of integer values*/
for $i = 1, ..., n_z$ **do**
  $\varsigma_i' \leftarrow \varsigma_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $u_1 = U(0,1); u_2 = U(0,1)$
  $p = 1 - \frac{\varsigma_i / n_z}{1 + \sqrt{1 + (\frac{\varsigma_i}{n_z})^2}}$
  $G_1 = \lfloor \frac{\ln(1-u_1)}{1-p} \rfloor; G_2 = \lfloor \frac{\ln(1-u_2)}{1-p} \rfloor;$
  $z_i' = T^z_{[z_i^{min}, z_i^{max}]}(z_i + G_1 - G_2)$
end
/*Mutation of discrete values*/
$p' = 1/[1 + \frac{1-p}{p} * \exp(-\tau_l * N(0,1))]$
for $i \in \{1, ..., n_d\}$ **do**
  if $U(0,1) < p_i'$ **then**
    $d_i' \leftarrow$ uniformly randomly value from D_i
  end
end

Algorithm 1. Mutation operator in MIES

evaluation just we mentioned above. For those expensive optimization problems, it is helpful to use metamodel to predict fitness values to reduce computation time. Typical metamodels include polynomial models, kriging models, neural networks and support vector machines [7]. For MIES, an additional challenge for constructing metamodels is that the model should be able to deal with multiple types of variables, unlike in most meta-model assisted EAs, where fitness functions of continuous variables only are involved. In [8], Li *et al.* has adopted an RBFN as metamodel for approximating functions having different types of variables.

Measuring distance between individuals is necessary when RBFN is employed for fitness estimation, because the activation function in the hidden layer involves calculating the distance between individuals. However the distance metric used for RBFN is based on one single type of variables in general. Therefore, how to measure the distance between mixed-integer individuals is crucial.

A straightforward way is to use different metrics for different variable types, and then combine these different metrics to obtain the distance between individuals. In [8], Li *et al* adopted the Euclidean distance for continuous variables, Manhattan distance for integer variables, and an overlap metric for nominal discrete variables. Then HEOM (Heterogeneous Euclidean-Overlap Metric) is used to combine different metrics. HEOM [14] is a heterogeneous metric which uses different attribute distance functions on different kinds of attributes and takes the square root of the sum of the various distances.

The formal statement of the distance between two individuals is described as follows [8]:

$$\Delta_x(x, x') = \sqrt{\Delta_r(r, r') + \Delta_z(z, z') + \Delta_d(d, d')} \tag{2}$$

$$\begin{gathered} \Delta_r(r, r') = \sum_{i=1}^{n_r} (r_i - r'_i)^2; \quad \Delta_z(z, z') = \sum_{i=1}^{n_z} |z_i - z'_i|; \\ \Delta_d(d, d') = \sum_{i=1}^{n_d} I(d_i \neq d'_i), \quad \text{with} \quad I(true) = 1, I(false) = 0. \end{gathered} \tag{3}$$

This modified RBFN was used as a metamodel to assist MIES, and successfully accelerated MIES on test problems as well as on the parameter optimization of an IVUS (intravascular ultrasound) image analysis feature detector, which is a real-world optimization problem [8]. Li *et al.* put forward an RBFN assisted MIES algorithm as described in algorithm 2 except the line 6 .

3 RBFN Based on Kendall Rank Correlation Coefficient

3.1 Kendall Rank Correlation Coefficient as a Distance Metric

Although it has been successful for the modified RBFN to predict fitness functions for MIES, much room remains for improvement. For instance, the distance metric may introduce bias into variables of different types. Suppose there are two continuous variables. One of the variables value ranges between 0 and 10, while the other variable value is between 0 to one million. We can judge that the latter is dominating in the calculated distance. It means that the range of a variable can bias its importance in the distance metric, which is however, not reasonable since all the variables should be equally important.

In this paper, we propose a new rank-based distance metric for RBFN to improve the performance of MIES assisted by RBFN. We first introduce Kendall Rank Correlation Coefficient before we provide the details about the new distance metric.

Kendall Rank Correlation Coefficient [10] (Kendall RCC) was proposed to evaluate the degree of similarity between two vectors. Suppose there are two random vectors $X(x_1, x_2, ..., x_n)$ and $Y(y_1, y_2, ..., y_n)$. And (x_1, y_1), (x_2, y_2),..., (x_n, y_n) are defined as a joint variable of X and Y. To measure the degree of similarity between the joint variables, concordant and discordant are introduced.

$$(x_i, y_i), (x_j, y_j) \text{is} \begin{cases} c & \text{if } (x_i > y_i \text{ and } x_j > y_j) \text{ or } (x_i < y_i \text{ and } x_j < y_j) \\ d & \text{if } (x_i > y_i \text{ and } x_j < y_j) \text{ or } (x_i < y_i \text{ and } x_j > y_j) \end{cases} \tag{4}$$

Here, c denotes the concordant and d discordant.

There are n joint variables and $\frac{1}{2}n(n-1)$ pairs of joint variable. For all the $\frac{1}{2}n(n-1)$ pairs of joint variables, we can obtain the number of concordant pairs c_{num} and the number of discordant pairs d_{num} .

The Kendall RCC τ is defined by:

$$\tau = \frac{(c_{num} - d_{num})}{\frac{1}{2}n(n-1)}. \tag{5}$$

The range of coefficient is from -1 to 1 because of the denominator is the number of the pairs of the joint variable. $\tau = 1$ if the relative ranks of two vectors are totally same; $\tau = -1$, if one of the relative ranks is the inverse of the other; $\tau = 0$, while the relative ranks of two vectors are fully independent.

Here, we propose a new distance metric for the mixed-integer RBFN base on RCC. In this paper, the Euclidean distance has been adopted for continuous variables, Manhattan metric for integer variables, and overlap metric for nominal discrete variables. Similarly, HEOM is adopted for combing different metrics.

$$\Delta_x(x, x') = \sqrt{\Delta_r(r, r') + \Delta_z(z, z') + \Delta_d(d, d')} \tag{6}$$

Here,

$$\begin{gathered} \Delta_r(r, r') = \sum_{i=1}^{n_r} w_{r_i}(r_i - r'_i)^2; \quad \Delta_z(z, z') = \sum_{i=1}^{n_z} w_{z_i}|z_i - z'_i|; \\ \Delta_d(d, d') = \sum_{i=1}^{n_d} w_{d_i} I(d_i \neq d'_i), \quad \text{with} I(true) = 1, I(false) = 0. \end{gathered} \tag{7}$$

3.2 RBFN Base on RCC Assisted MIES

How to determine the weight for each variable is of big importance, since the negative effect caused by the different ranges of variables needs to be avoided, whilst the relative ranks among individuals need to be ensured as much as possible. Before training RBFN, there are K^+ individuals whose true fitness values are known. Thus, the true fitness values of these individuals, and the variables of individuals are known. In determining the weights of variables, the main idea is that the variable that has a higher degree of similarity with true fitness values will be assigned a bigger weight.

Suppose an individual has m variables, and there are n individuals for training the RBFN. $X_1(f_1; a_{11}, a_{21}, ..., a_{m1})$, $X_2(f_2; a_{12}, a_{22}, ..., a_{m2})$,... $X_n(f_n; a_{1n}, a_{2n}, ..., a_{mn})$.

Here, X_i denotes the ith individual for training; f_i is the true fitness value of the ith individual; a_{ji} denotes the value of the jth variable of the ith individual. The order of the $m + 1$ vectors are $(f_1, f_2, ..., f_n)$, $(a_{11}, a_{12}, ..., a_{1n})$,..., $(a_{m1}, a_{m2}, ..., a_{mn})$. Here, $(f_1, f_2, ..., f_n)$ is a vector combined by all the true fitness values of n individuals. $(a_{j1}, a_{j2}, ..., a_{jn})$ is a vector combined by all the values of ith variable of n individuals. To determine the weight of variable a_j, we can obtain τ_i between two vectors $(f_1, f_2, ..., f_n)$ and $(a_{j1}, a_{j2}, ..., a_{jn})$. $w_i = \tau_i$.

The difference between RBFN assisted MIES and RCC-RBFN assisted MIES is that, before training RBFN in each time, we use the K^+ training individuals to compute the weights of the variables. The distance metric between two individuals can then be determined, once the weights of the variables are obtained. The RCC-RBFN assisted MIES as below.

$t \leftarrow 0$
Initialize population P_t of K^+, including μ, individuals randomly generated within the individuals space $\mathbb{I}$
Evaluate the P_t and insert results to database D
while *Termination criteria not fulfilled* **do**
  Compute the weights for variables
  Train RBFN based onK^+ latest evaluations
  Generate the λ^+ offspring
  Predict fitness of λ^+ offspring
  Select the best λ individuals out of λ^+ offspring
  Evaluate λ selected individuals by using original fitness function, and insert results to database D
  Select the μ best individuals for P^{t+1} from λ offspring
  $t \leftarrow t + 1$
end

Algorithm 2. Main Loop of RCC-RBFN Assisted MIES

4 Experimental Study

In order to assess the efficacy of RCC-RBCN, we applied RCC-RBFN assisted MIES to a number of test functions. Comparison of the performance of RCC-RBFN assisted MIES and RBFN assisted MIES has been conducted.

4.1 Test Functions

Four mixed integer optimization problems, denotes as $f_1 - f_4$, are chosen as test functions. f_1 is a mixed-integer sphere function chosen from [8], f_2 is a weighted sphere function, f_4 is a modified step function, f_2 - f_4 are chosen from [3]. The minimum of these test functions are all 0. The detailed information of the fitness functions can be found in [3, 8]. The test functions are described below. In these four test functions, we set $n_r = n_d = n_z = 5$. That is, the dimension of the test functions are 15. For all test functions, $r_i \in [0, 1000]$ $(1 \leq i \leq n_r)$, $z_i \in [0, 1000]$ $(1 \leq i \leq n_z)$, $d_i \in \{0, 1, ..., 9\}$ $(1 \leq i \leq n_d)$.

$$f_1(r, z, d) = \sum_{i=1}^{n_r} r_i^2 + \sum_{i=1}^{n_z} z_i^2 + \sum_{i=1}^{n_d} d_i^2 \rightarrow min \tag{8}$$

$$f_2(r, z, d) = \sum_{i=1}^{n_r} i r_i^2 + \sum_{i=1}^{n_z} i z_i^2 + \sum_{i=1}^{n_d} i d_i^2 \rightarrow min \tag{9}$$

$$f_3(r, z, d) = \sum_{i=1}^{n} (\sum_{j=1}^{i} (r_j + z_j + d_j))^2 \rightarrow min \tag{10}$$

$$f_4(r, z, d) = \sum_{i=1}^{n_r} \lfloor r_i \rfloor^2 + \sum_{i=1}^{n_z} (z_i \text{ div } 10)^2 + \sum_{i=1}^{n_d} (d_i \mod 2)^2 \rightarrow min \tag{11}$$

4.2 Configuration

In this paper, we adopt overlapping-generation model $(\mu+\lambda)$-MIES, also known as the plus strategy. It is a selection mechanism where the parents in each generation compete with offspring for survival rather than directly die off. $(\mu+\lambda)$-MIES converges faster than (μ, λ)-MIES, but may result in premature convergence. Since all test functions considered here are unimodal, the $(\mu + \lambda)$ strategy is adopted.

We adopted Gaussian function as the radial basis function for RBFN [15]. The standard deviation σ is set to $\sigma = \frac{d_{max}}{\sqrt{2m_1}}$. Here, m_1 is the number of the center and d_{max} is the maximum distance between individuals.

$$G(x, x_i) = \exp(-\frac{1}{2\sigma_i^2}\|x - x_i\|^2) \tag{12}$$

We set $\mu = 4$, $\lambda = 10$, $\lambda^+ = 36$ and $K = 64$. The average convergence histories of each algorithm on the $15 - D$ test functions f_1 - f_4 are shown in Figure 1.

4.3 Result and Analysis

To compare the performance among MIES, RCC-RBFN assisted MIES and RBFN assisted MIES, the profiles of evolutionary optimization of the algorithms were showed in Figure 1. The maximum number of allowed fitness evaluations of both algorithms was set 5000. 100 independent runs on each algorithms were collected. As we can see from the Figure 1, RCC-RBFN assisted MIES has performed much better than MIES and RBFN assisted MIES in all test functions.

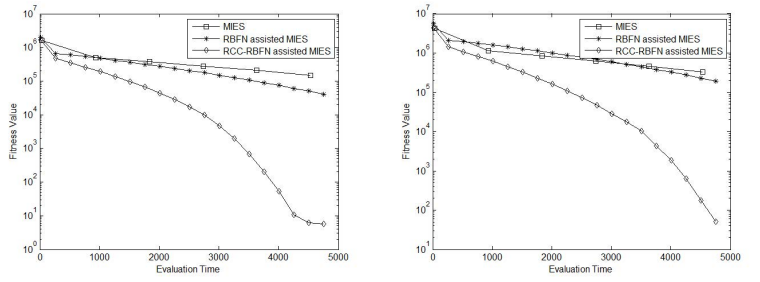
Table 1. Wilcoxon rank sum test

	standard MIES	RBFN assisted MIES	RCC-RBFN assisted MIES
$f1$	1.2154e+005±1.4646e+005 −	3.3215e+004±6.2000e+004 −	5.3681±26.8744
$f2$	2.7870e+005±4.2415e+005 −	1.5898e+005±2.3380e+005 −	20.6850±83.5361
$f3$	9.5717e+004±1.4152e+005−	4.0563e+004±8.2662e+004 −	0.5722±1.2554
$f4$	6.2876e+003±4.0548e+003−	2.4826e+003±2.6181e+003 −	162.2921±431.8199
−	4	4	
+	0	0	
≈	0	0	

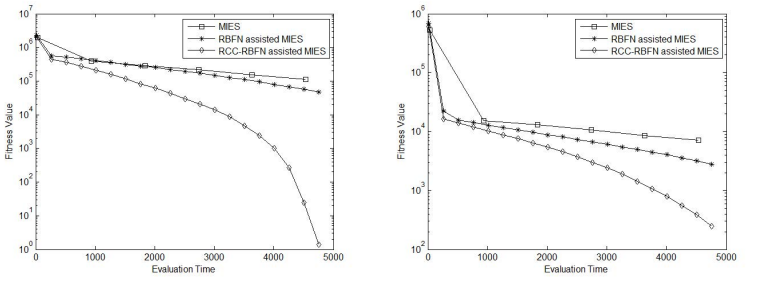
Note 1. +, − and ≈ denotes that the performance of the standard MIES and the RBFN assisted MIES is better than, worse than, or similar to RCC-RBFN assisted MIES according to the result of the Wilcoxon rank sum test.

In Table 1, the average and standard deviation of the best results are presented. The Wilcoxon rank-sum test with a significance level of 0.05 was used to compare the solutions of RBFN assisted MIES and RCC-RBFN assisted MIES. The results clearly indicate that RCC-RBFN assisted MIES wins in all four tests.

The results also imply that RCC-RBFN outperforms the standard RBFN in accerlating MIES. Thus, the new distance metric makes RBFN more effective when it is applied to assist MIES.



(a) test function 1 (b) test function 2



(c) test function 3 (d) test function 4

Fig. 1. Test functions

5 Conclusion

In this paper, we have proposed a rank correlation coefficient based RBFN to improve the performance of RBFN assisted MIES. Our experimental results show that the RCC-RBFN assisted MIES performs much better than the MIES assistanted by a distance-based RBFN.

The present work is very preliminary in that it has been tested only on four test functions and the dimension of the function is relatively low. More research is needed to extend the proposed RBFN model for solving more complex and high-dimensional mixed integer optimization problems. It is also hoped that the proposed method can be verified in solving real-world problems.

Acknowledgement. This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant 61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16.

References

1. Beyer, H.G., Schwefel, H.P.: Evolution strategies–a comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
2. Bäck, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press on Demand (1996)
3. Emmerich, M., Grötzner, M., Groß, B., Schütz, M.: Mixed-integer evolution strategy for chemical plant optimization with simulators. In: Evolutionary Design and Manufacture, pp. 55–67. Springer (2000)
4. Emmerich, M., Grötzner, M., Schütz, M.: Design of graph-based evolutionary algorithms: A case study for chemical process networks. Evolutionary Computation 9(3), 329–354 (2001)
5. Li, R., Emmerich, M., Eggermont, J., Bovenkamp, E.G.: Mixed-integer optimization of coronary vessel image analysis using evolution strategies. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 1645–1652. ACM (2006)
6. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation 6(5), 481–494 (2002)
7. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing 9(1), 3–12 (2005)
8. Li, R., Emmerich, M.T., Eggermont, J., Bovenkamp, E.G., Back, T., Dijkstra, J., Reiber, J.H.: Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In: IEEE Congress on Evolutionary Computation, CEC 2008 (IEEE World Congress on Computational Intelligence), pp. 2764–2771. IEEE (2008)
9. Engelbrecht, A.P.: Computational intelligence: An introduction. Wiley (2007)
10. Kendall, M.G., et al.: Rank correlation methods. Rank Correlation Methods (1948)
11. Schwefel, H.P.: Evolution and optimum seeking. sixth-generation computer technology series (1995)
12. Schütz, M.: Eine evolutionsstrategie für gemischt-ganzzahlige optimierungsprobleme mit variabler dimension. Master's thesis, FB Informatik, University Dortmund, Germany (1994)
13. Rudolph, G.: An evolutionary algorithm for integer programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 139–148. Springer, Heidelberg (1994)
14. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. arXiv preprint cs/9701101 (1997)
15. Haykin, S., Network, N.: A comprehensive foundation. Neural Networks 2 (2004)

Semi-supervised Ranking via List-Wise Approach

Zhigao Miao and Ke Tang*

USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI), School of Computer Science and Technology University of Science and Technology
of China, Hefei 230027, China
mzg123@mail.ustc.edu.cn, ketang@ustc.edu.cn

Abstract. Semi-supervised ranking is a relatively new and important learning problem. Some semi-supervised learning to rank algorithms have been proposed by adapting existing pair-wise ranking algorithms into the semi-supervised setting. In contrast, there is comparatively little work in developing semi-supervised ranking algorithm based on the list-wise approach. In this paper, we proposed a semi-supervised extension of one of the state-of-the art list-wise ranking algorithm—AdaRank, for the task of learning to rank with partially labeled data. The proposed algorithm is a two-staged approach that combines the process of label propagation and a regularized version of AdaRank. The label propagation process is used to propagate the relevance labels from labeled instances to other unlabeled ones, so that more training data are available to learn the ranking function. The regularized version of AdaRank optimizes a novel performance measure, which incorporates both the usual Information Retrieval (IR) metrics and the smoothness of the ranking function. Experimental results on benchmark datasets show the advantage of our methods over existing pair-wise algorithms for the Information Retrieval measures.

Keywords: Learning to rank, Semi-supervised learning, List-wise approach, Label propagation.

1 Introduction

Ranking problem, aiming at inducing an ordering over objects, arises in a wide variety of domains, such as information retrieval [1], natural language processing [2], and computational biology [3]. Consequently, there has been a surge of research in the field of "learning to rank" [1], which aims at using labeled training data and machine learning methods to construct reliable ranking functions.

Progress has been made in developing different algorithms for the fully supervised ranking task. Roughly speaking, most of these methods fall into two categories: the pair-wise approach [4] and the list-wise approach [4]. More recently, partially due to the great success achieved by semi-supervised learning [9], semi-supervised ranking algorithms have attracted more and more research interests

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 376–383, 2013.

[5,8,10]. However, most of these works are obtained by extending existing pair-wise algorithms into the semi-supervised setting, there is comparatively little work in extending list-wise approach to semi-supervised ranking. Hence in this paper, we conduct research on extending list-wise approach to semi-supervised setting, which may result in algorithms outperform the pair-wise ones.

In this paper, we proposed an extension of AdaRank [11] for the task of semi-supervised ranking. The proposed algorithm is a two-staged approach that combines the label propagation technique and a regularized version of AdaRank. In the phase of label propagation, we treat the relevance labels of instances as class labels and propagate labels according to the manifold structure of training data within the same query. Then the algorithm run the modified version of AdaRank on the dataset augmented by unlabeled data with pseudo labels. The regularized measure offers a trade-off between performance measures based on pseudo labels and the smoothness of the learned ranking function, which can further enhance the reliability and generalization performance of our algorithm.

The rest of the paper is organized as follows: Section 2 provides a brief literature review to the related work. In Section 3 we describe our proposed semi-supervised ranking algorithm in details and we present the results of our evaluation in Section 4. Finally, in Section 5 we conclude the paper and give directions for future work.

2 Related Work

2.1 Semi-supervised Learning to Rank

In the semi-supervised ranking problem, a set of queries $Q = \{q_1, q_2, \ldots, q_m\}$ is given. Each query q_i is associated with a list of documents $\boldsymbol{d}_i = \{d_{i1}, d_{i2}, \ldots, d_{i,n(q_i)}\}$ and a list of labels $\boldsymbol{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{i,n(q_i)}\}$ where $y_{ij} \in \{r_1, r_2, \ldots, r_k\}$ is the relevance level of d_{ij} and y_{ij} is only available for the labeled items. Our task is to learn a function $f(x; w)$ with parameters w that ranks the documents for each query in the testing set.

Several methods have been developed for the task of semi-supervised ranking. For example, Amini et al. [5] proposed an adaption of supervised RankBoost [6] algorithm which builds the ranking function on the basis of two training sets: one labeled and one unlabeled. First, the algorithm uses an unsupervised method to initially assign relevance judgments to the unlabeled examples. Then an extended version of RankBoost algorithm is developed to produce a scoring function by optimizing the pair-wise loss function. Another line of work is the extension of Rank-SVM [7] within the semi-supervised regularization framework. Pan et al. [8] proposed Semi-RankSVM which is a combination of RankSVM and graph Laplacian regularization trick that can exploit the neighborhood manifold of the training data. The method adopts a large margin optimization approach like the traditional RankSVM, and minimizes the pair-wise loss together with a graph-based regularization term. Other extensions of ranking algorithms based on semi-supervised techniques, such as Co-Training [9], Self-Training [9], Multi-view Learning [9] are also possible [10].

The majority of these methods seek to learn by combining the pair-wise approach and certain semi-supervised technique. However, as noted by Cao et al. [4] there are some drawbacks in the pair-wise approach. The pair-wise approach defines loss functions on the level of document pairs and the documents in a query are viewed as equivalently important, which are not in accordance with the evaluation criteria such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). These evaluation metrics are defined on the query-level, and usually pay more attention to the higher ranked documents. Therefore, minimizing the loss functions in the pair-wise approach does not necessarily imply enhancing the ranking performances.

In the list-wise ranking approach, the list of retrieved documents for a query is viewed as an example in learning. Moreover, some existing state-of-the-art list-wise methods, such as AdaRank [11], can directly optimize performance measures of IR such as MAP and NDCG. Recent works has shown that the list-wise approach usually performs better than the pair-wise one. In the following, we give a simple visit to the AdaRank algorithm that will be used in our algorithm.

2.2 AdaRank

Recently, a number of authors have proposed conducting direct optimization of multivariate performance measures in ranking problems. AdaRank [11] is a list-wise ranking algorithm which iteratively optimizes an exponential loss function based on any of IR performance measures employing the boosting technique.

AdaRank takes document lists (queries) as examples and repeatedly constructs weak rankers on the basis of re-weighted training data, then linearly combines the weak rankers for making ranking predictions. At each round, AdaRank increases the weights of those queries that are not ranked well by the model created so far. As a result, the learning at the next round will be focused on the creation of a weak ranker that can focus on the ranking of those hard queries. When the iteration end, AdaRank outputs the ranking model by linearly combining the obtained weak rankers, where the combination coefficients are determined by the probability distributions on the query lists.

3 Our Method

The challenge of extending list-wise approach is to select a reasonable list-wise ranking algorithm and corresponding suitable semi-supervised techniques, then figure out ways to combine these two basic components. In the following, we first outline the general framework of our method, then describe the proposed algorithm, namely "LP-AdaRank", in details.

3.1 General Framework

When performing list-wise ranking algorithms, the labels for the whole dataset and the ground truth list for queries are needed as supervised information. In

order to incorporate unlabeled data in learning, we rely on a preceding semi-supervised algorithm to generate this information for training. Our proposed solution is a very simple two-staged method. First we treat the relevance levels of examples as their class labels and employing certain semi-supervised algorithm with probabilistic outputs to learn the class distribution within each query. The class distribution information allow us to formulate a sensible and stable way to get the two kinds of information. After this, we run list-wise ranking algorithms on the augmented training data. The general framework of our method is:

1. Semi-supervised classification for label distribution within each query.
2. Transforming the class probabilities into the supervised information.
3. Training the list-wise ranker on dataset for prediction.

Here we choose label propagation [12] and AdaRank [11] as such a pair. The label propagation process can effectively utilize the unlabeled data to proliferate the training set, and it is totally independent on the ranking algorithms. While AdaRank provides a flexible framework that shares the advantages of theoretical soundness, efficiency in training and high performance in testing.

3.2 Label Propagation

Label propagation [12] is a commonly used semi-supervised method which assumes that closer data points tend to have similar class labels, and propagates labels by employing the neighborhood manifold structure of data. Label propagation simulates a diffusion process that propagates the labels to neighboring edges leading eventually to labels for the whole set.

In our proposed algorithm we regard the relevance levels of examples as class labels, and run label propagation algorithm within each query. For convenience, we have displayed just a single query. Denote n as the total number of labeled and unlabeled examples for the query under consideration, C as the number of classes(or relevance levels). The neighborhood manifold structure of data can be naturally represented by an empirical graph where nodes represent the training data and edges represent similarities between them. These similarities are given by a weight matrix W:

$$w_{ij} = \begin{cases} 1 & \text{if } i \in N_K(j) \text{ or } j \in N_K(i) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $N_K(.)$ is the K nearest neighbors of relevant node. Normalize W to get transition probabilities matrix T, and define a $n \times C$ label matrix Y, whose ith row representing the label probabilities of node i. As proved in Zhu [12], the algorithm is guaranteed to converge regardless of the initialization of the rows of Y corresponding to unlabeled data points. The major steps of label propagation algorithm are as follows:

1. All nodes propagate labels for one step : $Y \leftarrow TY$.
2. Row-normalize Y to maintain the class probability interpretation.
3. Fix the labels on the labeled data. Repeat from step 1 until Y converges.

3.3 Our Algorithm

The proposed algorithm involves two steps: Label Propagation, and Ranking Function Learning. The label propagation process offers the labels and the ground-truth lists for the list-wise ranking algorithm. We confine the propagation within the documents returned by a single query. When the propagation process ends, the labels of unlabeled data are obtained by taking the class with maximal label probability as in Eq. (2). The conventional way to obtain the total order ground-truth list for query q_i is to sort examples by their relevance levels, while the ties are broken by randomization. However this simple method may result in unstable results. To overcome this disadvantage, we use the method in Li et al. [13] which take the distributions information of labels into account and give different weights for all possibilities. The ground-truth list for query q_i is obtained by sorting s_i in Eq. (3) by the descending order. Here $T(j)$ can be any increasing function of relevance level j. When $T(j) = j$, s_i can be explained as expected relevance level.

$$y_i^* = \arg\max_j Y_{ij} \tag{2}$$

$$s_i = \sum_{j=1}^{C} P(y_i^* = j)T(j) = \sum_{j=1}^{C} T(j)Y_{ij} \tag{3}$$

At the second stage, it is a natural idea to use the usual IR measure MAP or NDCG as the performance function for the construction of weak ranker in each iteration. However, the calculation of these performance measures needs the true label of instances, we only have pseudo labels for unlabeled data. Instead of treating the pseudo labels as true labels for unlabeled data and directly run the original AdaRank on the augmented dataset, we see that both the labeled information and smoothness of ranking function are critical to enhance the ranking accuracy and hence we should use both of them to define performance functions in learning. In our proposed performance function in Eq. (4), we can easily modify the AdaRank algorithm to use our proposed performance function in constructing weak rankers.

$$E(f,q) = \frac{1}{2}(NDCG(f,q) + \exp(-\lambda \sum_{i,j} w_{ij}||f(x_i) - f(x_j)||^2)) \qquad (\lambda > 0) \tag{4}$$

The parameter λ is a trade-off factor to control the importance of the two parts which can be selected by cross-validation routine. It is easy to check that proposed performance measure favors the model varying more smoothly over the data manifolds.

Algorithm 1. Our derived algorithm

Input: $S = f(q_i, \boldsymbol{d}_i, \boldsymbol{y}_i)_{i=1}^{m}$
Parameters: neighborhood size K, number of iterations T, trade-off factor λ.
• Run label propagation within each query and get labels for all data.
• Initialize $P_1(i) = 1/m$.
For $t = 1, \ldots, T$
• Create weak ranker h_t with weighted distribution P_t on training data.
• Choose α_t $\quad \alpha_t = \frac{1}{2} \ln \frac{\sum_{i=1}^{m} P_t(i)\{1+E(h_t,q_i)\}}{\sum_{i=1}^{m} P_t(i)\{1-E(h_t,q_i)\}}$
• Create f_t $\quad f_t(x) = \sum_{k=1}^{t} \alpha_k h_k(x)$
•Update P_{t+1} $\quad P_{t+1}(i) = \frac{\exp\{-E(f_t,q_i)\}}{\sum_{i=1}^{m} \exp\{-E(f_t,q_i)\}}$
Output ranking model : $f(x) = \sum_{i=1}^{T} \alpha_t h_t(x)$.

4 Experiment

4.1 Experiment Setup

In this section, we present an empirical evaluation of our semi-supervised ranking algorithm. The proposed algorithm LP-AdaRank was compared with two representative semi-supervised pair-wise ranking algorithm Semi-RankBoost [5], Semi-RankSVM [8] to validate our approach and to show the superiority of semi-supervised ranking algorithms based on the list-wise approach.

The experiments are conducted on MQ2008-semi(Million Query track) and OHSUMED datasets. The MQ2008-semi dataset [14] is public available benchmark data collections for research on semi-supervised learning to rank. There are about 2000 queries in this dataset. The OHSUMED dataset [15] consists of 106 queries with 25 features extracted from the on-line medical information database. For the OHSUMED dataset, we reduce the amount of available labeled data in the training set by forming a 10% random sample of all the labels for each query to imitate the case of limited labels. Both datasets were split in five folds with a 3:1:1 ratio for training, validation and testing. The results reported in this section are average results over multiple folds.

As to evaluate the performance of ranking models, we use Normalized Discounted Cumulative Gain (NDCG) as evaluation measures. Given a query q_i, the NDCG score at position n in the ranking list of documents can be calculated by the equation as follows. Here $r(j)$ is the rating of the jth document in the list and Z_n is the normalization constant.

$$NDCG@n = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log(1+j)} \tag{5}$$

4.2 Experiment Results

There are two learning parameters to be tuned in our algorithm. For simplicity, we have fixed the learning parameter to reasonable defaults: nearest neighborhood size $K = 10$. The trade-off factor λ in the performance measure varied from 0 to 1 with a step of 0.05, and the best value on the validation set was chosen. The averages result across the 5 folds are summarized in the following table.

The results show that a great improvement was made in terms of the NDCG measure for both of the two datsets. This indicates that our proposed algorithm has better performance than the two baselines, and justifies the benefits of semi-supervised ranking algorithms based on the list-wise approach. The list-wise approach thus appears as a promising direction for improving the performance of semi-supervised ranking tasks.

Table 1. NDCG@n MEASURES ON THE MQ2008-SEMI COLLECTION

Algorithm	NDCG@1	NDCG @3	NDCG @5	NDCG @ 7	NDCG @ 10
Semi-RankBoost	0.463	0.455	0.449	0.412	0.430
Semi-RankSVM	**0.495**	0.420	0.416	**0.413**	0.414
LP-AdaRank	0.484	**0.462**	**0.451**	0.405	**0.447**

Table 2. NDCG@n MEASURES ON THE OHSUMED COLLECTION

Algorithm	NDCG@1	NDCG @3	NDCG @5	NDCG @ 7	NDCG @ 10
Semi-RankBoost	0.501	**0.483**	0.431	0.459	0.426
Semi-RankSVM	0.482	0.471	0.402	0.453	0.434
LP-AdaRank	**0.528**	0.477	**0.464**	**0.472**	**0.465**

5 Conclusion and Future Work

In this paper, we present a semi-supervised ranking algorithm that combines the label propagation technique and a regularized version of AdaRank. The label propagation process, together with the regularized performance measures, have largely enhanced the reliabity and generalization performance of the ranking algorithm when labeled data is scarce. Experiments show that the proposed algorithm outperforms existing pair-wise ranking algorithms. For future work, we plan to investigate different alternatives for the supervised list-wise ranking algorithm and the semi-supervised techniques, and at the same time to conduct further empirical evaluations of the proposed algorithm.

Acknowledgements. This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant 61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Natural Science Foundation of Anhui Province under Grant 1108085J16.

References

1. Liu, T.Y.: Learning to Rank for Information Retrieval. Journal of Foundations and Trends in Information Retrieval, 225–331 (2002)
2. Collins, M., Koo, T.: Discriminative Reranking for Natural Language Parsing. Journal of Computational Linguistics, 25–70 (2005)
3. Weston, J., Kuang, R., Leslie, C., Noble, W.S.: Protein Ranking by Semi-supervised Network Propagation. Journal of BMC Bioinformatics (2006)
4. Cao, Z., Qin, T., Liu, T., Li, H.: Learning to Rank: from Pair-wise Approach to List-wise Approach. In: Proceedings of the 24th International Conference on Machine Learning, pp. 129–136 (2007)
5. Amini, M.R., Truong, T.V., Gutte, C.: A Boosting Algorithm for Learning Bipartite Ranking Functions with Partially Labeled Data. In: Proceedings of the 31st Annual International Conference on Research and Development in Information Retrieval, pp. 99–106 (2008)
6. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research, 933–969 (2003)
7. Cao, Y., Xu, J., Liu, T., Li, H., Huang, Y., Hon, H.: Adapting Ranking SVM to Document Retrieval. In: Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval, pp. 186–193 (2006)
8. Pan, Z.B., You, X., Chen, H., Tao, D.C., Pang, B.C.: Generalization Performance of Magnitude-preserving Semi-supervised Ranking with Graph-based Regularization. Journal of Information Sciences, 284–296 (2013)
9. Zhu, X.J.: Semi-supervised Learning Literature Survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison (2005)
10. Truong, V., Amini, M., Gallinari, P.: A Self-training Method for Learning to Rank with Unlabeled Data. In: Proceedings of the 16th European Symposium on Artificial Neural Networks, pp. 22–24 (2009)
11. Xu, J., Li, H.: AdaRank: A Boosting Algorithm for Information Retrieval. In: Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval, pp. 391–398 (2007)
12. Zhu, X.J., Ghahramani, Z.: Learning from Labeled and Unlabeled Data with Label Propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon (2002)
13. Li, P., Burges, J.C., Wu, Q.: McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (2007)
14. Liu, T.Y., Xu, J., Qin, T., Xiong, W.Y.: LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. Journal of Information Retrieval, 346–374 (2010)
15. Hersh, W., Buckley, C., Leone, T.J., Hickam, D.: OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In: Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval, pp. 192–201 (1994)

Gaussian Process for Transfer Learning through Minimum Encoding*

Hao Shao[1], Rui Xu[2], and Feng Tao[3]

[1] School of WTO Research and Education,
Shanghai University of International Business and Economics
[2] School of Computer Science and Technology,
University of Science and Technology of China
[3] School of Business, East China University of Science and Technology

Abstract. In real applications, labeled instances are often deficient which makes the classification problem on the target task difficult. To solve this problem, transfer learning techniques are introduced to make use of existing knowledge from the source data sets to the target data set. However, due to the discrepancy of distributions between tasks, directly transferring knowledge will possibly lead to degenerated performance which is also called *negative trasnfer*. In this paper, we adopted the Gaussian process to alleviate this problem by directly evaluating the distribution differences, with the parameter-free Minimum Description Length Principle (MDLP) for encoding. The proposed method inherits the good property of solid theoretical foundation as well as noise-tolerance. Extensive experiments results show the effectiveness of our method.

Keywords: Transfer Learning, Gaussian Process, MDLP.

1 Introduction

Transfer learning [13] provides a solution for real applications to induce models on a target task given a large amount of auxiliary data. Take the four universities example into consideration, given a set of data sets of different universities and colleges, we can find useful information from them to help further classification on a data set of a new university (or a college) into several categories such as students, faculties, stuffs and so on. Assume that data sets of two universities, denoted by U_1, U_2, and two colleges, denoted by C_1, C_2, are in the source domain, we try to classify instance in a new college, say C_t, with a few labeled data.

Directly applying the latent functions or hypothesis drawn from the source domain generally may not lead to satisfactory performance on the target domain,

* This work was supported by Humanity and Social Science Youth foundation of Ministry of Education of China (No. 13YJC630126), the Fundamental Research Funds for the Central Universities (No.WK0110000032), the NSFC (No.71171184/71201059/71201151), the Funds for Creative Research Group of China (No. 70821001) and the NSFC major program (No.71090401/71090400).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 384–391, 2013.

which is regarded as negative transfer. Some of the information from the source domain may probably hinder the performance if the distributions between the source domain and the target domain are too dissimilar [10]. Although there exists extensive research works on transfer learning, as stated in the survey paper [13], how to avoid negative transfer is an important issue but yet not be studied intensively. In the example, if we directly use the hypothesi on U_1 to C_t, the error rate may be very high due to the different distributions underlying the two data sets. To avoid negative transfer, it is reasonable to consider the similarities between a source task and a target task, and several works have been done to find the similarities between tasks [1,12,2,3].

In multi-task learning, the similarities between tasks can be evaluated by distances between instances or the similarity between distributions. It is reasonable because all the tasks are treated as symmetric. There are enough instances in each task for us to find the underlying distributions. However, difficulties are encountered when we try to deal with transfer learning. In transfer learning setting, instances in the target task is often inadequate, generally it is difficult to find the distribution. Methods to measure the similarities for multi-task learning can not be directly adopted and the negative transfer tends to happen. Furthermore, labeling instances in the target task is expensive and time consuming.

In this paper, we tackle this problem from a two-level point of view, and try to directly solve the negative transfer problem by adopting the Gaussian process. Our main idea to calculate the task-level similarity is based on the purpose to use only auxiliary information in the source domain. The intuition is that it is difficult to find the ground-truth distribution on the target task, so we do not try to perform operations on the target data but only to use the labeled data for testing the "hypotheses". Therefore, an MDLP (Minimum Description Length Principle) [11] based method is adopted, because it can evaluate models on data sets and also parameter-free and robust to noise. For the instance-level similarity, we design a kernel functions by using the obtained information from the task-level similarity. We call our algorithm the GPTL (Gaussian Process for Transfer Learning).

2 Problem Setting and Preliminaries

2.1 Gaussian Process for Classification

A Gaussian process is a stochastic process and a collection of random variables. All these variables in a finite number have joint Gaussian distributions. We can interpret in that, the Gaussian distribution is over vectors, and the Gaussian process is over functions. We have the following definition that, a function f is distributed as a Gaussian process with mean function m and covariance function k, which is also denoted as a "kernel".

$$f \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{1}$$

Due to the space limit, we will introduce GP in more detail in the extension of this manuscript.

2.2 Preliminaries for Encoding

A data set D consists of n examples $d_1, d_2, \ldots, d_n$. Each example d_i is described with attributes $a_1, \ldots, a_m$ as an attribute value vector $(a_{i1}, a_{i2}, \ldots, a_{im})$ of length m and belongs to one of M classes, of which labels are represented by $c_1, c_2, \ldots, c_M$. A classifier h is a function which outputs a class label given an attribute value vector. We call the process of learning a classifier from a data set classification. A classifier with a simple interpretation and a high classification accuracy is most preferable.

The MDLP may be viewed as a principle for avoiding overfitting, i.e., it is a means to balance the simplicity of a classifier and its goodness-of-fit to the data [4,9]. As a principle for preferring a classifier in classification, it is stated in MDLP that the best classifier h_{best} among the ones h that can be learned from D is given as follows.

$$h_{\text{best}} = \arg\min_{h} \left(-\log P(h) - \log P(D|h)\right) \tag{2}$$

Due to the space limit, we neglect the detailed explanation of the theoretical background of GPTL, readers can refer to [5,7,6] or the extended version of this manuscript. We provide the coding methods here considering encoding a binary string of length a which consists of b binary 1s and $(a-b)$ binary 0s [4,9]. In the sender and the receiver problem, the receiver is assumed to know the length a. An obvious method is to send the number b of binary 1s with the code length $\log(a+1)$ then specify the positions of binary 1s with the code length $\log \binom{a}{b}$ [4,9]. We hereafter call this method a binary coefficient method and denote the required code length by $\Theta(a, b)$ as follows.

$$\Theta(a, b) \equiv \log(a+1) + \log \binom{a}{b}$$

Now we consider a problem of sending an integer a under the assumption that $a = b$ is most likely and the occurrence probability $P(i)$ of $a = i$ is given by $P(b)\phi^{|b-i|}$, where ϕ is a constant given by the user.

We denote the length $-\log\left[P(b)\phi^{|b-a|}\right]$ required to send a given b by $\Lambda(a, b)$ and it can be easily obtained as

$$\Lambda(a, b) = \log\left[3 - \left(\frac{1}{2}\right)^{b}\right] + |b-a| \tag{3}$$

3 The GPTL Algorithm

3.1 Arrange Related Tasks

For the source tasks $S_1, S_2, ..., S_{\text{k}}$, we can easily obtain the latent function $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$ underlying the source data sets based on the Gaussian process. We

denote the $\mathbf{f}_i$ as the "hypothesis" in the MDLP. As it is difficult to find the true distribution underlying the target data set given inadequate instances, our idea is to use existing latent functions and fit each $\mathbf{f}_i$ to the target data set to evaluate the similarities. MDLP has advantages to do such jobs.

In order to find the source data set which has the most similar distributions with the target data set, we need to find the one in $\mathbf{f}_i$ ($i = 1$ to k) which can maximize the probability $P(\mathbf{f}_i|D_t)$. It turns out to be the maximization of:

$$\arg\max_{\mathbf{f}_i} \frac{P(D_t|\mathbf{f}_i)P(\mathbf{f}_i)}{P(D_t)} \tag{4}$$

In the evaluation procedure, $P(D_t)$ is neglected as a constant. By negative encoding with MDLP, this could be calculated as

$$L_i \equiv -\log P(D_t|\mathbf{f}_i) - \log P(\mathbf{f}_i) = -\log \prod_i P(d_i^t|\mathbf{f}_i) - \log P(\mathbf{f}_i) \tag{5}$$

where d_i^t represents the i-th instance in D_t.

This could be regarded as the sender and receiver problem, and evaluated by MDLP. Then the key issue is to design a delicate coding method. Note that, a simple way to encode the first part of formula (2) is to specify the wrong predictions of the class labels in T based on $\mathbf{f}_i$. The second part can be regarded as the complexity of the latent function $\mathbf{f}_i$. This also coincides with the basic format of the refined MDL with one part as the stochastic complexity of the data related to the model and the other part as the parametric complexity.

To encode $-\log P(\mathbf{f}_i)$, since each $\mathbf{f}_i$ is assumed to follow a Gaussian distribution as $\mathbf{f}_i \sim \mathcal{N}(0, K)$, we only need to calculate the code length of the kernel matrix K (K is a matrix with each entry K_{ij} as a kernel function $k(x_i, x_j)$). Commonly, the squared-exponential covariance function is adopted as a kernel in Gaussian process:

$$k(x_i, x_j) = \sigma_f^2 \exp(-\frac{1}{2l^2}(x_i - x_j)^2) \tag{6}$$

Then we only need to code the two parameters σ_f and l. A direct way is to code the two real values using $\Lambda(\sigma_f, 0)$ and $\Lambda(l, 0)$. Notice that, even with limited information from labeled data in the target task, we could obtain roughly a function $\mathbf{f}_t$ as similar to the true distribution of T. Let the corresponding parameters in the kernel function of $\mathbf{f}_t$ be as σ'_f and l'. Therefore, we can use $\Lambda(\sigma_f, \sigma'_f)$ and $\Lambda(l, l')$ to send the two values under the assumption that $\sigma_f = \sigma'_f$ and $l = l'$ are most likely. Then the code length should be:

$$-\log P(\mathbf{f}_i) = \Lambda(\sigma_f, \sigma'_f) + \Lambda(l, l') \tag{7}$$

Then we obtain the code length L_i between each S_i and T. The source task with the shortest code length is regarded to "best fit" the target data, thus the

distribution is more similar to the one in the target task. Then the source data sets are arranged in ascending order on the code length with the target data set T.

3.2 The Instance Level Similarities

To represent the similarity between two instances from different tasks, we are motivated to construct a kernel function by using the code length which is regarded as the similarity measurements between tasks. Based on an existed kernel $k(x_i, x_j)$, we design an extended one $\tilde{k}(x_i, x_j)$ as follows:

$$\tilde{k}(x_i, x_j) = exp(\frac{1}{1 + exp(-L_i)})k(x_i, x_j) \tag{8}$$

It is easy to prove that $\tilde{k}(x_i, x_j)$ is also a kernel if $exp(\frac{1}{1+\exp(-L_i)})$ is regarded as a positive constant.

The intuition behind formula (8) is that, the kernel function should take both the task similarity and instance similarity together. Note that, when instances are from the same task, the coefficient $exp(\frac{1}{1+\exp(-L_i)})$ is set to be 1 and thus the kernel function takes the original format.

4 Experiments

We perform experiments on the data sets from the UCI repository[1] and the Text data sets. The three data sets in the UCI repository used in the experiments are *mushroom* and *splice*. We adopted a pre-processing method [14,16] on them to fit the transfer learning scenario. For the Text data sets, we choose 20NewsGroup data sets[2].

The *mushroom* data set has 8124 examples with 22 attributes in each example and one binary class label. The *splice* data set has 3190 examples with 60 attributes in each example and one binary class label. We adopt the same strategy in [14] and [16] to split each data set into two.

The number of examples in the source task is set to be 1000 which is the same as in [16]. We investigated the influence of the number of instances in the target data set, and the noise level in the target data set. The noise is added by reversing the correct class labels of the examples in the training data sets.

We follow the splitting strategy on 20Newsgroups data sets as [14]. Three data sets are chosen which are *rec* vs *talk*, *rec* vs *sci* and *sci* vs *talk*. For example, in *rec* vs *talk* data set, all the positive instances are from the category *rec*, while negative ones are from the category *talk*. The instances in the source domain and the target domain are selected based on the subcategories. In the experiments, each of the target tasks in the three data sets are chosen as the single target

[1] `http://archive.ics.uci.edu/ml/`
[2] `http://people.csail.mit.edu/jrennie/20Newsgroups`

task, and the training data in the three data sets are all chosen as the source tasks. In such a way, $\mathbb{S}$ contains 3 different tasks as S_1, S_2 and S_3 and we test our algorithm in this transfer learning setting.

Our GPTL is compared with the COITL [16] and TrAdaBoost [14], Active Transfer (AT) [15] and k-NN with $k = 3$ as well as the basic SVM. The hyperplanes are obtained by C-SVC with polynomial kernel, which are considered effective to data with a large number of features and without class noise.

For the UCI data sets, each of them has one source task and one target task, GPTL is thus used only in the instant level to select useful parts. For the Text data sets, both the two levels are examined. We mainly test two factors in the experiments. One is the different number of instances in the target task. We set $|T|$ equals to 50 and 100, respectively. The other is the noise level in the target task from 0% to 15%.

Table 1 and 2 provide the results on the *mushroom* data set and the *splice* data set, respectively. Generally speaking, the error rates go up with the noise level increases. If the target domain has more labeled instances, for example, $|T| = 100$, the accuracy is obviously better. From Table 1 we observed that GPTL is better than other methods in most circumstances. However, because the *mushroom* data set is well organized, for all the methods, it is easier to find the underlying hypotheses. That is why in some cases our algorithm is outperformed by the state-of-the-art methods. In Table 2, in this larger and more complex data set, our method is the best among all the methods, and the improvements are much larger than that in the mushroom data set.

Table 1. Results on *mushrom* data set

		Percentage of noise on T					
		0%	3%	6%	9%	12%	15%
$\|T\|$=50	SVM	0.087	0.12	0.146	0.173	0.19	0.207
	GPTL	**0.082**	**0.11**	**0.132**	**0.147**	**0.163**	0.165
	TrAdaBoost	0.158	0.159	0.173	0.191	0.168	0.195
	KNN	0.117	0.125	0.153	0.147	0.167	0.163
	COITL	0.132	0.144	0.146	0.161	0.168	0.159
	AT	0.156	0.196	0.177	0.16	0.185	**0.142**
$\|T\|$=100	SVM	0.067	**0.052**	0.111	0.129	0.167	0.198
	GPTL	**0.061**	0.074	**0.079**	**0.103**	**0.105**	0.136
	TrAdaBoost	0.145	0.143	0.158	0.178	0.167	0.166
	KNN	0.081	0.084	0.104	0.12	0.147	0.159
	COITL	0.103	0.08	0.087	0.121	0.11	**0.112**
	AT	0.2	0.189	0.177	0.199	0.184	0.178

Table 3, Table 4 and Table 5 provide the results on *rec* vs *talk*, *rec* vs *sci*, and *sci* vs *talk*, respectively, with $|T|$ equals to 50 and 100. From these table, it can be seen that our method is obviously better than other methods, even under noise conditions. It is a proof of the robustness of our method given only a few labeled instances in the target domain. We also notice that, when the *sci* vs *talk* data set is used as the target task, the error rates for all the methods are slightly higher than that of the other two data sets. The possible reason is that the discrepancy of distributions between the source domain and the target domain is large.

Table 2. Results on *splice* data set

		Percentage of noise on T					
		0%	3%	6%	9%	12%	15%
$\|T\|$=50	SVM	0.302	0.321	0.354	0.368	0.391	0.346
	GPTL	**0.2**	**0.211**	**0.235**	**0.243**	**0.296**	**0.314**
	TrAdaBoost	0.343	0.296	0.318	0.323	0.314	0.39
	KNN	0.375	0.354	0.347	0.379	0.398	0.415
	COITL	0.385	0.401	0.388	0.422	0.394	0.436
	AT	0.468	0.47	0.441	0.469	0.48	0.478
$\|T\|$=100	SVM	0.232	0.23	0.276	0.293	0.309	0.322
	GPTL	**0.19**	**0.207**	**0.232**	**0.23**	**0.249**	**0.255**
	TrAdaBoost	0.234	0.3	0.302	0.267	0.294	0.299
	KNN	0.331	0.349	0.352	0.377	0.367	0.396
	COITL	0.319	0.339	0.327	0.362	0.367	0.374
	AT	0.472	0.458	0.474	0.484	0.465	0.477

Table 3. Results on *rec* vs *talk* as the target task

		Percentage of noise on T					
		0%	3%	6%	9%	12%	15%
$\|T\|$=50	SVM	0.154	0.174	0.189	0.228	0.227	0.23
	GPTL	**0.135**	**0.138**	**0.165**	**0.185**	**0.203**	**0.219**
	TrAdaBoost	0.236	0.234	0.275	0.309	0.32	0.321
	KNN	0.207	0.255	0.237	0.256	0.244	0.305
	COITL	0.206	0.255	0.229	0.25	0.241	0.294
	AT	0.472	0.364	0.38	0.388	0.488	0.49
$\|T\|$=100	SVM	**0.088**	0.166	0.177	0.269	0.285	0.295
	GPTL	0.105	**0.144**	**0.16**	**0.179**	**0.24**	**0.258**
	TrAdaBoost	0.265	0.283	0.338	0.348	0.355	0.338
	KNN	0.23	0.248	0.267	0.277	0.281	0.283
	COITL	0.224	0.245	0.257	0.267	0.278	0.283
	AT	0.406	0.363	0.492	0.477	0.315	0.517

Table 4. Results on *rec* vs *sci* as the target task

		Percentage of noise on T					
		0%	3%	6%	9%	12%	15%
$\|T\|$=50	SVM	0.163	0.177	0.187	0.181	0.223	0.24
	GPTL	**0.133**	**0.145**	**0.152**	**0.176**	**0.215**	**0.21**
	TrAdaBoost	0.266	0.289	0.292	0.353	0.351	0.375
	KNN	0.245	0.224	0.255	0.246	0.261	0.3
	COITL	0.243	0.236	0.259	0.255	0.268	0.311
	AT	0.41	0.37	0.44	0.325	0.354	0.419
$\|T\|$=100	SVM	0.139	0.152	0.153	0.174	0.21	0.213
	GPTL	**0.136**	**0.142**	**0.15**	**0.169**	**0.19**	**0.202**
	TrAdaBoost	0.24	0.23	0.24	0.228	0.267	0.284
	KNN	0.216	0.185	0.192	0.182	0.212	0.231
	COITL	0.206	0.2	0.194	0.183	0.22	0.231
	AT	0.433	0.316	0.399	0.359	0.315	0.438

Table 5. Results on *sci* vs *talk* as the target task

		Percentage of noise on T					
		0%	3%	6%	9%	12%	15%
$\|T\|$=50	SVM	0.183	0.195	0.223	0.24	0.307	0.364
	GPTL	**0.175**	**0.191**	**0.203**	**0.22**	**0.25**	**0.256**
	TrAdaBoost	0.301	0.282	0.317	0.372	0.372	0.409
	KNN	0.285	0.305	0.301	0.327	0.307	0.385
	COITL	0.286	0.303	0.318	0.327	0.31	0.385
	AT	0.368	0.425	0.446	0.382	0.366	0.371
$\|T\|$=100	SVM	0.221	0.235	0.239	0.264	0.329	0.342
	GPTL	**0.165**	**0.173**	**0.18**	0.196	0.21	**0.215**
	TrAdaBoost	0.195	0.219	0.255	0.244	0.213	0.23
	KNN	0.184	0.185	0.196	0.194	0.194	0.233
	COITL	0.172	0.184	0.188	**0.188**	**0.192**	0.234
	AT	0.352	0.399	0.323	0.327	0.418	0.362

5 Conclusion

This paper proposed a transfer learning algorithm based on Gaussian process. By directly evaluating the distribution differences between tasks, we alleviate the problem of negative transfer. MDLP was also adopted to balance both the simplicity of the hypothesis and the goodness-of-fit to the data. In the experiments, our method was proved to outperform other methods and also be robust to noise.

References

1. Argyriou, A., Maurer, A., Pontil, M.: An algorithm for transfer learning in a heterogeneous environment. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 71–85. Springer, Heidelberg (2008)
2. Bakker, B., Heskes, T.: Task clustering and gating for bayesian multitask learning. Journal of Machine Learning Research 4, 83–99 (2003)
3. Cao, B., Pan, S.J., Yang, Q.: Adaptive Transfer Learning. In: AAAI 2010 (2010)
4. Wallace, C., Patrick, J.: Coding Decision Trees. Machine Learning 11(1), 7–22 (1993)
5. Shao, H., Tong, B., Suzuki, E.: Compact Coding for Hyperplane Classifiers in Heterogeneous Environment. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 207–222. Springer, Heidelberg (2011)
6. Shao, H., Tong, B., Suzuki, E.: Extended MDL Principle for Feature-based Inductive Transfer Learning. Knowledge and Information Systems 35(2), 365–389 (2013)
7. Shao, H., Suzuki, E.: Feature-based Inductive Transfer Learning through Minimum Encoding. In: SDM 2011, pp. 259–270 (2011)
8. Shao, H., Tong, B., Suzuki, E.: Query by Committee in a Heterogeneous Environment. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS, vol. 7713, pp. 186–198. Springer, Heidelberg (2012)
9. Quinlan, J.R., Rivest, R.L.: Inferring Decision Trees Using the Minimum Description Length Principle. Information and Computation 80(3), 227–248 (1989)
10. Rosenstein, M.T., Marx, Z., Kaelbling, L.P.: To Transfer or Not To Transfer. In: NIPS 2005 Workshop on Transfer Learning (2005)
11. Grünwald, P.D.: The Minimum Description Length Principle. MIT Press, Cambridge (2007)
12. Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 567–580. Springer, Heidelberg (2003)
13. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2009)
14. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for Transfer Learning. In: ICML 2007, pp. 193–200 (2007)
15. Shi, X., Fan, W., Ren, J.: Actively Transfer Domain Knowledge. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 342–357. Springer, Heidelberg (2008)
16. Shi, Y., Lan, Z., Liu, W., Bi, W.: Extended Semi-supervised Learning Methods for Inductive Transfer Learning. In: ICDM 2009, pp. 483–492 (2009)

Kernel Based Manifold Learning for Complex Industry Fault Detection

Jian Cheng and Yi-nan Guo

CUMT-IoT Perception Mine Research Center,
School of Information and Electrical Engineering,
China University of mining and Technology, Xuzhou, 221116, China
chengjian@cumt.edu.cn

Abstract. Accurate and rapid fault detection based on the data from industry process is very important for the process control. This paper introduces a new multivariate statistical process control approach for fault detection using kernel method based manifold learning algorithm combining T^2 statistic. The proposed approach is effective in the fault detection, which has two stages. Stage I: a kernel method based locally linear embedding is employed to extract the nonlinear features, preserve local structure and reduce dimensionality of the multivariate input data, and a new low-dimensional embedding method is developed to solve the "out-of-sample" problem. Stage II: the fault detection is performed by T^2 statistic with control limits derived from the eigenanalysis of the kernel matrix in the Hilbert feature space. In this study, the method is applied for the fault detection of the benchmark Tennessee Eastman (TE) challenge process. The proposed method has been compared with conventional methods in terms of performances such as detection accuracy, detection delay and false alarm rate. It is demonstrated that the proposed method outperformed the others in fault detection on TE process.

Keywords: Manifold learning, Kernel method, Fault detection, TE process.

1 Introduction

Generally, process monitoring consists of measuring and controlling several process variables simultaneously[1], which is increasing difficult to detect and diagnose the fault states if multiple process variables exhibit outliers or deviations at the same time. To overcome this disadvantage, multivariate quality control methods has been employed by monitoring the interactions of several process variables and determining hidden factors using dimensionality reduction methods[2], such as principal component analysis (PCA), dynamic principal component analysis (DPCA), Fisher Discriminant Analysis (FDA) and partial least squares (PLS)[3][4]. Subsequently, applying multivariate statistics to the low-dimensional data representations produced by these methods, faults can be

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 392–400, 2013.

detected with greater accurate. However, for some complex situations in industrial field with particularly nonlinear characteristics, there are poor performances due to the assumption that the process data are linear for these methods.

This shortcoming of linear methods has led to the development of nonlinear methods. Then a series of nonlinear PCA method based on neural networks has been developed[5][6]. However, it is difficult to determine structure of neural network and neural network training consumes a lot of time. Another method is based on kernel method, for example, kernel PCA (KPCA)[7] and one-class support vector machines (1-class SVM)[20]. Although these methods can deal with the nonlinear problem, they all do not consider the potential structure of the input data.

For nonlinear dimensionality reduction, Manifold learning is a perfect tool which can discover and retain the structure of high dimensional data sets for a better understanding of the data. Several different manifold learning algorithms have been developed to perform dimensionality reduction, such as Isomap[9], locally linear embedding (LLE)[10], Laplacian eigenmaps (LE)[11], and Maximum Variance Unfolding (MVU)[12], etc.

The LLE algorithm is considered one of effective manifold learning algorithms for dimensionality reduction, and has been used to solve various problems in information processing, pattern recognition and data mining[13][14]. LLE algorithm computes a different local quantity, and calculates the best coefficients to approximate each point by a weighted linear combination of its neighbors, and then tries to find a set of low-dimensional points, which can be linearly approximated by its neighbors with the same coefficients that have been determined from high-dimensional points. However, LLE may fail to deliver good performance when the data structure is nonlinear. Moreover, it faces the difficulty of how to implement the map on new testing data points.

Because the the sensor data collected from process is typically nonlinear, high-dimensional and generally correlated, it is necessary to develop a new method to tackle the fore-mentioned drawbacks of traditional methods. Recently, there has been great interest in developing low dimensional representations through kernel based methods[15][16][17]. These methods can efficiently discover the nonlinear structure of data and evaluate the map on out-of-sample data. In this paper, the proposed method includes two stages. Firstly, we present a kernel method based LLE algorithm (KLLE) to reduce the dimensionality of the input data and obtain a low-dimensional embedding data. In the second stage, we adopt Hotelling's T^2 statistic chart[1] in the embedding data to determine the control limit and detect the fault state. This method can not only deal with the nonlinear problem, but also preserve the potential structure in the input data points.

The remainder of the paper is organized as follows. Section 2 presents kernel method based manifold learning. In Section 3, we introduces T^2 statistic in Hilbert feature space for fault detection. We experimentally evaluate the performance of our proposed method using TE process data in section 4. Section 5 concludes this paper.

2 Kernel Method Based Manifold Learning

2.1 Kernel Method Based LLE

Let $X = \{x_1, x_2, \cdots, x_n\}$ be a given data set of n points in $\mathbb{R}^m$, sampled from a $d-$dimensional manifold ($d \leq m$). LLE constructs a dataset $Y = \{y_1, y_2, \cdots, y_n\} \subseteq \mathbb{R}^d$. There are the details of LLE in[10]. With the kernel trick, suppose that the input space X is mapped into a Hilbert feature space $\mathbb{H}$ through a nonlinear mapping function $\phi : \mathbb{R}^m \to \mathbb{H}$[15]. In Hilbert feature space $\mathbb{H}$, the nearest neighborhood of $\phi(x_i)$ is $\{\phi(x_i^j), j = 1, \cdots, l\}$, where l is the number of nearest neighbors of $\phi(x_i)$. As in LLE, it is worth to note that the Euclidean distance between two data points in the Hilbert feature space can be computed according to

$$\| \phi(x_i) - \phi(x_j) \|^2 = k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j). \tag{1}$$

Then, in Hilbert feature space, the first step in LLE is to determine the neighbor set for $\phi(x_i)$ and to learn the local linear structure of the neighbor set by solving

$$\min \| \phi(x_i) - \sum_{j=1}^{l} w_{ij} \phi(x_i^j) \|^2, \tag{2}$$

The weights w_{ij} can be computed by minimize Equation 2 with the constraints $\sum_{j=1}^{l} w_{ij} = 1$. We can get the following Lagrange formulation

$$L(W_i) = W_i^T C_i W_i - \lambda(W_i^T \mathbf{e} - 1), \tag{3}$$

where C_i is the local kernel matrix of $\phi(x_i)$ in $\mathbb{H}$, and

$$\begin{aligned} C_i(j,k) &= (\phi(x_i) - \phi(x_i^j))^T (\phi(x_i) - \phi(x_i^k)) \\ &= k(x_i, x_i) - k(x_i, x_i^k) - k(x_i, x_i^j) + k(x_i^j, x_i^k). \end{aligned} \tag{4}$$

Equation 3 which is subjected to $W_i^T \mathbf{e} = 1$ ($\mathbf{e}$ is a one dimension vector consisting of ones) has the closed form solution $W_i = C_i^{-1}\mathbf{e}/(\mathbf{e}^T C_i^{-1}\mathbf{e})$ [18]. C_i is a positive definite matrix, the eigen-decomposition of C_i is of form $C_i = U^T \Lambda U$, then

$$W_i = (U^T \Lambda^{-1} U\mathbf{e})/(\mathbf{e}^T U^T \Lambda^{-1} U\mathbf{e}). \tag{5}$$

Hence, the reconstruction weights W are computed by C's eigenvalues and eigenvectors.

2.2 KLLE Embedding

In Hilbert feature space, we suppose that the embedding Y can be given by $Y = \Gamma^T \phi(X)$, where Γ is a linear transformation matrix and $y_i = \Gamma^T \phi(x_i) \in \mathbb{R}^d$.

Now, we turn to the problem of finding a transformation matrix Γ in Hilbert feature space $\mathbb{H}$. The best low-dimensional embedding Y can be computed by

$$\begin{aligned}\sum_i \parallel y_i - \sum_j w_{ij} y_j \parallel^2 &= \parallel Y(I-W) \parallel^2 \\ &= tr[\Gamma^T \phi(X) \widetilde{M} \phi(X)^T \Gamma],\end{aligned} \tag{6}$$

where $Y = [y_1, y_2, \cdots, y_n] = \Gamma^T \phi(X)$ and $\widetilde{M} = (I-W)(I-W)^T$. Since each column of Γ should lie in the span of $\phi(x_1), \phi(x_2), \cdots, \phi(x_n)$, we can write

$$\Gamma = [\sum_i a_1(i)\phi(x_i), \sum_i a_2(i)\phi(x_i), \cdots, \sum_i a_d(i)\phi(x_i)] = \phi(X)A, \tag{7}$$

where $a_k(i), k = 1, \cdots, d$, denotes the ith entry of the coefficient vector a_k, and $A = [a_1, a_2, \cdots, a_d] \in \mathbb{R}^{n \times d}$. Substituting Equation 7 to Equation 6, we can obtain

$$\min_{\Gamma} \sum_i \parallel y_i - \sum_j w_{ij} y_j \parallel^2 = \min_A tr(A^T K \widetilde{M} K A), \tag{8}$$

where $K = \phi(X)^T \phi(X)$ is a $(n \times n)$ symmetric kernel matrix whose entries are $K(i,j) = k(x_i, x_j)$. Similarly, $\frac{1}{n} YY^T = I$ becomes to $\frac{1}{n} A^T K K A = I$.

Finally, the constrained minimization problem above is converted to the following generalized eigenvalue problem

$$K \widetilde{M} K a = \lambda K K a. \tag{9}$$

And the matrix A is determined by the eigenvectors corresponding to the bottom d nonzero eigenvalues of Equation 9. Once A is obtained, for any data point x in high dimensional space $\mathbb{R}^m$, it can be mapped to a low dimensional space point $y \in \mathbb{R}^d$ by

$$y = \Gamma^T \phi(x) = A^T [k(x_1, x), k(x_2, x), \cdots, k(x_n, x)]^T, \tag{10}$$

and we can obtain another form as follow

$$y_j = \sum_i^n a_j^i k(x_i, x), \quad j = 1, \cdots, d, \tag{11}$$

where y_j is the jth entry of embedding coordinate y.

3 T^2 Statistic for Fault Detection

Using the KLLE, we can collect the embedding coordinates $Y = [y_1, y_2, \cdots, y_n]^T \in \mathbb{R}^{n \times d}$. A measure of the variation within the KLLE is given by Hotelling's T^2 statistic[2]. T^2 is the sum of the normalized squared scores, and is defined as

$$T^2 = Y \Lambda^{-1} Y^T, \tag{12}$$

where Λ^{-1} is the diagonal matrix of the inverse of the eigenvalues associated with the embedding coordinates obtained by the generalized eigen decomposition of Equation 9.

Given a level of significance α, appropriate threshold value for the T^2 statistic can be determined automatically by applying the probability distributions. In this study, the control threshold for T^2 is obtained using the $F-$distribution

$$T^2_{d,n,\alpha} \sim \frac{d(n-1)}{n-d} F_{d,n-d,\alpha}, \tag{13}$$

where n is the number of samples and d is the number of embedding dimensions.

4 Experiments and Results

4.1 Tennessee Eastman Process and Data Acquisition

The Tennessee Eastman (TE) plant-wide industrial process control problem was proposed by Downs and Vogel, described in detail in[19][3]. The dataset includes all the manipulated and measured variables, except the agitation speed of the reactor's stirrer for a total of $m = 52$ observation variables(The agitation speed was not included because it was not manipulated).

The simulation time for each run is 96 hours. The first 48 hours are operated under normal operating conditions, the fault is induced after 48 hours. A sampling interval of 3 minutes is used to collect the simulated data for the training (normal state) and testing data (fault state). The total number of observations generated for each run is 1920 samples. The normal operating training data consists of 960 samples. When no faults induced, we can obtain other 960 normal state samples as validation data. The control threshold of T^2 statistic can be set based on this validation data in the next section. In the following section, the performance of KLLE is compared with PCA, DPCA, KPCA and 1-class SVM for TE process.

4.2 Fault Detection of the TE Process

We compared the fault detection performance of kernel based LLE (KLLE) monitoring method with that of PCA[3], DPCA[4], KPCA and 1-class SVM[8] based monitoring method. Fault detection performance was evaluated by detection accuracy and false alarm rate of each method. In accordance with the works by Mahadevan et al.[8] and Chiang et al.[3], the fault is indicated only when six consecutive statistic values exceed the threshold and the detection delay of a monitoring chart is defined as the time gap between the introducing of fault and the statistic value exceeding its upper control threshold for the first time. Since it is unfair to compare detection accuracy and detection delays of all methods when they have different false alarm rate, in computing above indices, the control threshold for each monitoring statistic in each method was adjusted to the 10th highest value of the normal operating validation data. In this way, the adjusted threshold corresponds to the 99% confidence limit.

Table 1. The detection accuracy for the testing data

Model	Fault1	Fault2	Fault4	Fault5	Fault6	Fault7	Fault8	Fault10	Fault11
PCA+T^2	99.1	97.3	-	23.5	98.9	91.3	96.3	33.3	20.5
DPCA+T^2	99.4	98.1	-	24.2	98.7	84.1	97.2	42.0	19.9
KPCA+T^2	**100**	99.3	83.7	29.8	**100**	**100**	97.6	80.8	81.3
1-class SVM	99.8	98.6	**99.6**	**100**	**100**	**100**	97.9	87.6	69.8
KLLE+T^2	**100**	**99.6**	96.2	**100**	**100**	**100**	**99.4**	**96.6**	**87.5**

Model	Fault12	Fault13	Fault14	Fault16	Fault17	Fault18	Fault19	Fault20	Fault21
PCA+T^2	97.9	94.0	84.3	16.7	75.1	88.7	-	29.7	26.4
DPCA+T^2	99.0	95.1	93.9	21.7	76.0	88.9	-	35.6	35.6
KPCA+T^2	98.4	95.5	**100**	77.6	95.2	91.3	75.7	72.2	81.7
1-class SVM	**99.9**	95.5	**100**	89.8	**95.3**	90.0	83.9	90.0	52.8
KLLE+T^2	98.8	**97.5**	**100**	**89.9**	87.9	**92.9**	**84.4**	**96.6**	**81.9**

"-" denotes that the fault cannot be detected.

According to Chiang et al.[3] and Russell et al.[4] detection accuracies for faults 3, 4, 9, 15 and 19 were very low because there was no observable change in their means, variances or the higher moments. Hence it has been considered that these faults are unobservable and cannot be detected by any traditional statistical technologies. In the work of Mahadevan et al.[8], 1-class SVM method was also not able to detect faults 3, 9 and 15 efficiently, but can detect faults 4 and 19. Therefore, in this paper, our comparative experiments have not included faults 3, 9 and 15, although we can monitor all faults in TE process by the proposed method.

The fault detection accuracies are calculated and tabulated in Table 1. The maximum fault detection value obtained for each of the faults has been highlighted in bold face. As expected, DPCA statistic have performed better than that of PCA for most faults, this indicates the potential advantage of taking serial correlation into account by DPCA when developing fault detecting procedures. However, the fault detection accuracies of PCA and KPCA are still low, and the average of detection accuracies of PCA and DPCA over all faults are only 67.1% and 69.3% respectively, because they are linear method which cannot capture and model the nonlinear of TE process.

Table 2. The average of detection accuracy (ADA), detection delays (ADD) and false alarm rate (AFAR)

Model	ADA(%)	ADD	AFAR(%)
PCA+T^2	67.1	102.0	1.60
DPCA+T^2	69.3	84.4	**0.99**
KPCA+T^2	87.0	26.2	1.47
1-class SVM	91.7	32.8	1.36
KLLE+T^2	**94.9**	**1.4**	1.25

Obviously, the fault detection accuracies based on KPCA, 1-class SVM and KLLE are much higher than PCA and DPCA due to their nonlinearity via kernel method. For some faults such as fault 1, 2, 6, 7 and 14, these kernel based

method have similar detection accuracy, but 1-class SVM and KLLE are better than KPCA for most fault states demonstrated by average detection accuracy in Table 2. It is also observed that KLLE performs better than 1-class SVM for most of the faults except fault 4, 12 and 17, and the average of detection accuracies of 1-class SVM and KLLE are up to 91.7% and 94.9% in Table 2 respectively. In the case of fault 5, KLLE easily distinguishes the faulty data from the normal operating data. However, KPCA monitoring chart cannot detect it.

The false alarm rate of all the methods are summarized in Table 2. It can be seen that the DPCA method is much better than the rest of the models. Nevertheless, the false alarm rates of all the methods are well within acceptable limits. However, it should be noted that KLLE performs much better than other methods in its reduced detection delay and increased fault detection accuracy.

Mentioned above, the faults 3, 9 and 15 are unobservable from the testing data according to PCA, DPCA, KPCA and 1-class SVM. Although these three fault states have no observed change in the mean, the variance and the higher order variance, they can induce the change of data structure captured by KLLE. Therefore, it is encouraging that these three faults can be successfully detected by our proposed method. The detection accuracies of the faults 3, 9 and 15 are 77.7%, 90.7% and 73.0%, and the false alarm rates are 2.50%, 1.35% and 1.88% respectively.

From the results above, it is obviously shown that KLLE performs excellently on fault detection. Two facts demonstrate this capability. On the one hand, in nonlinear structures data set, KLLE preserves intrinsic properties more than the PCA, DPCA and KPCA. On the other hand, KLLE preserves the data structure more than 1-class SVM, and we also found that the time consumption of KLLE is smaller than 1-class SVM.

5 Conclusion

The application of machine learning to data mining and analysis in area of TE process is rapidly gaining interest in the community. In this paper, we presented a new effective approach to detect all the fault states in TE process. Different from conventional monitoring methods, KLLE can not only capture and model the nonlinearity, but also preserve the potential structure in the data. Moreover, the proposed method can get a implicit mapping relation between the original data space and the low-dimensional feature space by using kernel trick and linear projection, which make it possible to monitor online. For the fault detection of TE process, KLLE performed much better than other conventional monitoring methods such as PCA, DPCA, KPCA and 1-class SVM. Compared with these technologies, the proposed method could detect the fault with an increased detection accuracy and a considerable reduction in detection delay, and could generalize well to all the faults of TE process. Meanwhile, the false alarm rates were also within the acceptable limits, thus making it more useful and feasible for industrial online application.

Acknowledgments. This work is partially supported by the Fundamental Research Funds for the Central Universities (Grant No. 2011QNB24) and Jiangsu Natural Science Foundation (Grant No. BK2010183). The authors would like to thank the anonymous reviewers and editors for their help comments and suggestions.

References

1. Montgomery, D.C.: Introduction to Statistical Quality Control. John Wiley & Sons (2005)
2. Yang, K., Trewn, J.: Multivariate Statistical Methods in Quality Management. McGraw-Hill Professional (2004)
3. Chiang, L.H., Russell, E.L., Braatz, R.D.: Fault Detection and Diagnosis in Industrial Systems. Springer (2001)
4. Russell, E.L., Chiang, L.H., Braatz, R.D.: Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems 51(1), 81–93 (2000)
5. Jia, F., Martin, E.B., Morris, A.J.: Nonlinear principal components analysis with application to process fault detection. Journal of Systems Science 31(5), 1473–1487 (2001)
6. Dong, D., McAvoy, T.J.: Nonlinear principal component analysis based on principal curves and neural networks. Computers and Chemical Engineering 20(1), 65–78 (1996)
7. Schölkpof, B., Smola, A., Müller, K.-R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5), 1299–1319 (1998)
8. Mahadevan, S., Shah, S.L.: Fault detection and diagnosis in process data using one-class support vector machines. Journal of Process Control 19(10), 1627–1639 (2009)
9. Tenenbaum, J.B., Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290, 2319–2323 (2000)
10. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding, vol. 290, pp. 2323–2326 (2000)
11. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation 15, 1373–1396 (2003)
12. Weinberger, K.Q., Saul, L.K.: An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding. In: Proceedings of the 27th National Conference on Artificial Intelligence, pp. 1683–1686. AAAI Press (2001)
13. Elgammal, A.M., Lee, C.S.: Separating style and content on a nonlinear manifold. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern, pp. 478–485. IEEE Press (2004)
14. Mekuz, N., Bauckhage, C., Tsotsos, J.K.: Face recognition with weighted locally linear embedding. In: Proceedings of The Second Canadian Conference on Computer and Robot Vision, pp. 290–296. IEEE Press (2005)
15. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
16. Choi, H., Choi, S.: Robust Kernel Isomap. Pattern Recognition 40(3), 853–862 (2007)
17. Yu, X., Wang, X., Liu, B.: Supervised kernel neighborhood preserving projections for radar target recognition. Signal Processing 88(9), 2335–2339 (2008)

18. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. The Journal of Machine Learning Research 4, 119–155 (2003)
19. Downs, J.J., Vogel, E.F.: A Plant-wide Industrial Process Control Problem. Computers & Chemical Engineering 17(3), 245–255 (1993)
20. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, R.: Estimating the support of a high-dimensional distribution. Nerual Computation 13(7), 1443–1471 (2001)

An Estimation of Distribution Algorithm for the 3D Bin Packing Problem with Various Bin Sizes

Yaxiong Cai[1], Huaping Chen[1], Rui Xu[1,⋆], Hao Shao[2], and Xueping Li[3]

[1] School of Computer Science and Technology,
University of Science and Technology of China,
230026 Hefei, China
[2] School of WTO Research & Education,
Shanghai University of International Business and Economics,
200336 Shanghai, China
[3] Department of Industrial and Information Engineering,
University of Tennessee,
Knoxville, TN 37996-0700, USA

Abstract. The 3D bin packing problem (3DBPP) is a practical problem modeled from modern industry application such as container ship loading and plane cargo management. Unlike traditional bin packing problem where all bins are of the same size, this paper investigates a more general type of 3DBPP with bins of various sizes. We proposed a modified univariate marginal distribution algorithm (UMDA) for solving the problem. A packing strategy derived from a deepest bottom left packing method was employed. The modified UMDA was experimentally compared with CPLEX and a genetic algorithm (GA) approach. The experimental study showed that the proposed algorithm performed better than GA and CPLEX for large-scale instances.

Keywords: Optimization, Bin packing problem, Estimation of distribution algorithm, Genetic algorithm.

1 Introduction

Bin packing, or container loading is very common in industrial applications, such as cutting of foam rubber in arm-chair production, container ship loading,

⋆ This work was supported in part by the National Natural Science Foundation of China (No.71171184/61175065/71201151), the Funds for Creative Research Group of China (No. 70821001), the NSFC major program (No.71090401/71090400), the China Postdoctoral Science Foundation (No.2011M501067/2013T60628), the Fundamental Research Funds for the Central Universities (No.WK0110000032), the Program for New Century Excellent Talents in University (No.NCET-12-0512), the Science and Technological Fund of Anhui Province for Outstanding Youth (No.1108085J16), the Humanity and Social Science Youth foundation of Ministry of Education of China (No.13YJC630126), and the Cai Yuanpei Program (No.27927VE).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 401–408, 2013.

plane cargo management, and warehouse management. With a suitable packing strategy, it is able to reduce cost of transportation or fully utilize the manufacture materials. Another important benefit of minimizing the wasted space in a container is that densely packing container tends to protect the products better and decreases the chance of breakage during shipment. A lot of algorithms and packing strategy had been proposed for the traditional three-dimensional bin packing problem (3DBPP) where all bins are of the same size. However, in practical application we often face the situations that there are bins of different sizes to be used.

This paper investigates a unique variant of 3DBPP. In the investigated problem, there are more than one bin to be used and the bin sizes can differ from each other. Six item orientations are considered. The 3DBPP with various bin sizes contains the classical bin packing problems, where all the bins are of the same size. Unlike the traditional 3DBPP where only the number of bins used needs to be considered, the different sizes of bins pose a greater challenge in providing quality solution. The flexibility of item orientations also expands the search space significantly and hence increases the difficulty of finding optimal solution. This paper proposed an estimation of distribution algorithm for the investigated problem and experimentally tested it. The results showed that the proposed algorithm is a good optimizer for 3dBPP with various bin sizes.

The rest of the paper is structured as follows: The relevant literature is given in Section 2. Section 3 defines the problem precisely and presents the problem formulation. Section 4 presents an estimation of distribution algorithm for the problem. Section 5 conducts the experimental study by comparing the proposed algorithm with CPLEX and a genetic algorithm. Section 6 concludes the paper.

2 Literature Review

The solution techniques in literature for bin packing problems can be classified into three groups. The first group contains algorithms that use mathematical approach and try to reach an exact optimal solution. Martello et al.[2] discussed the lower bound and presented an exact algorithm to pack items into a single bin. Hifi et al.[1] introduced a mixed-integer linear programming formulation (MILP1) for bin packing problem. Heuristics and approximate algorithms belong to the second type. Lim et al.[3] did a comprehensive review on the heuristics that is used for bin packing problem. Most of them are wall or layer-building of the traditional 3DBPP to pack a selected subset of items into a single bin pursuing least waste space. Meta-heuristics such as genetic algorithm belong to the third type. Lodi et al.[4] presented a general Tabu Search technique for the solution of two- and three-dimensional bin packing problems. Crainic et al.[5] presented a new tabu search-based two-level which separates the search for the optimal number of bins from the optimization of the accommodation of items within bins, resulting into a more flexible procedure than the existing ones. Wang et al.[6] presented a hybrid genetic algorithm and a new crossover method for 3DBPP.

3 Problem Description and Formulation

The investigated problem comes from a real world application, where a transportation company tries to make full use of their various bins to pack the cartons which is almost all rectangular. Their goal is to minimize the sum of all the waste space in the bins that are used. Wasted space of a bin is denoted as the volume of the bin minus the sum of the volume of all cartons in the bin.

3.1 Problem Description

There are total m cartons and N boxes, all of which are rectangular. The length, width and height of carton i are denoted as l_i, w_i, h_i, while the length, width and height of bin i is denoted as L_i, W_i, H_i. All the cartons need to be packed into the bins under three constrains.

(1) Each carton lies completely in the bin and does not penetrate the bin's boundary surface. (2) No two cartons in the same bin can overlap with each other. (3) Each carton has only six rotations. Let the length of the bin be X-axis, the width of the bin be the Y-axis and the height of the bin be the Z-axis. Let (l, w, h) means the carton is placed in the way that length of the carton is paralleled to X-axis, width is paralleled to Y-axis and height is paralleled to Z-axis. The six rotation is (l, w, h), (l, h, w), (w, l, h), (w, h, l), (h, w, l), (h, l, w).

3.2 Problem Formulation

In this section, a CPLEX model based on the model introduced by Wu et al.[7] is presented.

Introduced variables:

N: total number of bins

m: total number of cartons

M: a large enough number, an acceptable value is $\sum_{i=1}^{m}(l_i + h_i + w_i) + \sum_{i=1}^{N}(L_i + H_i + W_i)$

(l_i, w_i, h_i): length, width and height of carton i

(L_i, W_i, H_i): length, width and height of bin i

s_{ij} : binary variables, $s_{ij} = 1$ means carton i is in bin j

n_i: binary variables, $n_i = 1$ means bin i is used

x_i, y_i, z_i: continuous variables, coordinate of the front-left-bottom corner point of carton i

$lx_i, ly_i, lz_i, wx_i, wy_i, wz_i, hx_i, hy_i, hz_i$: binary variables, indicating the rotation of carton i. For example, $ly_i = 1$ means that length of carton i is paralleled to X-axis.

$left_{ij}, right_{ij}, front_{ij}, behind_{ij}, below_{ij}, above_{ij}$: binary variables, indicating the relative position between two cartons. For example, $behind_{ij} = 1$ means carton i is behind carton j.

constrains:

for all i in $[1...m]$, k in $[i + 1...m]$

$$x_i + l_i * lx_i + w_i * wx_i + h_i * hx_i \leq x_k + (1 - left_{ik}) * M \tag{1}$$

$$x_k + l_k * lx_k + w_k * wx_k + h_k * hx_k \leq x_i + (1 - right_{ik}) * M \tag{2}$$

$$y_i + l_i * ly_i + w_i * wy_i + h_i * hy_i \leq y_k + (1 - front_{ik}) * M \tag{3}$$

$$y_k + l_k * ly_k + w_k * wy_k + h_k * hy_k \leq y_i + (1 - behind_{ik}) * M \tag{4}$$

$$z_i + l_i * lz_i + w_i * wz_i + h_i * hz_i \leq z_k + (1 - below_{ik}) * M \tag{5}$$

$$z_k + l_k * lz_k + w_k * wz_k + h_k * hz_k \leq z_i + (1 - above_{ik}) * M \tag{6}$$

for all i in $[1...m]$, k in $[i + 1...m]$, j in $[1...N]$

$$left_{ik} + right_{ik} + front_{ik} + behind_{ik} + below_{ik} + above_{ik} \leq s_{ij} + s_{kj} - 1 \tag{7}$$

for all i in $[1...m]$, k in $[i + 1...m]$, j in $[1...N]$

$$\sum_{j=1}^{N} s_{ij} = 1 \tag{8}$$

$$lx_i + ly_i + lz_i = 1 \tag{9}$$

$$wx_i + wy_i + wz_i = 1 \tag{10}$$

$$hx_i + hy_i + hz_i = 1 \tag{11}$$

$$lx_i + wx_i + hx_i = 1 \tag{12}$$

$$ly_i + wy_i + hy_i = 1 \tag{13}$$

for all i in $[1...m]$, j in $[1...N]$

$$x_i + l_i * lx_i + w_i * wx_i + h_i * hx_i \leq L_j + (1 - s_{ij}) * M \tag{14}$$

$$y_i + l_i * ly_i + w_i * wy_i + h_i * hy_i \leq W_j + (1 - s_{ij}) * M \tag{15}$$

$$z_i + l_i * lz_i + w_i * wz_i + h_i * hz_i \leq H_j + (1 - s_{ij}) * M \tag{16}$$

for all j in $[1...N]$

$$\sum_{j=1}^{m} s_i j \leq M * n_j \tag{17}$$

Objective:

$$min\{\sum_{j=1}^{N} L_j * W_j * H_j * n_j - \sum_{i=1}^{m} l_j * w_j * h_j\} \tag{18}$$

Constraints (1) to (6) ensure that any two cartons do not overlap with each other. Constraint (7) ensures that we will only check no overlap constrain when two cartons are placed in the same bin. Constraint (8) ensures that each carton will be placed in exactly one bin. Constraints (9) to (13) ensure that each carton's length, width, and height are paralleled to exactly one axis. Constraints (14) to (16) ensure that none of the cartons penetrate the bin's boundary surface. Constrains (17) ensures that a bin is considered used if any cartons is placed in it.

4 The Proposed EDA

In this section, a modified univariate marginal distribution algorithm (UMDA) for the investigated problem is introduced. This section is structure as: The first subsection introduces the representation strategy for our algorithm. The second subsection describes the detailed steps of the modified UMDA.

4.1 Representation

Representation is a strategy that translates a solution into sequences of numbers or find a suitable solution from number sequences. A good representation can help EDA achieve high quality solution easily. A population has P individuals. We use three number sequences to represent an individual. They are $\{bin_1, ..., bin_N\}$ which is the order of bins to be selected, $\{rotation_1,...,rotation_m\}$ which is the rotation ways of each carton, $\{carton_1,...,carton_m\}$ which is the order of cartons to be packed. The array $\{bin_1, ..., bin_N\}$ is a permutation of $\{1, 2, ..., N\}$. The array $\{carton_1... , carton_m\}$ is a permutation of $\{1, 2, ..., m\}$. $rotation_i$ is a integer that can be 0, 1, 2, 3, 4 or 5, which means the correspond carton is placed in the dimension of (l, w, h), (l, h, w), (w, l, h), (w, h, l), (h, w, l), (h, l, w) respectively.

Our packing strategy which finds a solution from number sequences is as follows: The cartons are placed into the bins one by one according to the sequence cartons. Carton i uses a rotation of $rotation_i$. The cartons are first tried to be placed into the bin bin_1, and if there is no space for any remaining cartons in bin bin_1, then the bin bin_2 is used. This process will continue, until either all the cartons are placed into the bins or the bins are used up and some cartons are left. In the first case, i.e., all the cartons are placed into the bins, the fitness is calculated as the total volume of the used bins minus the total volume of the cartons; in the second case, i.e., the bins are used up and some cartons are left, the fitness is set to be as the total volume of the used bins. The packing strategy for the single bin we use is similar as the "deepest bottom left first" method[6].

4.2 Generating New Population

The UMDA uses a probability model generated from the existing population to sample new populations. If the same method is applied for 3DBPP, we will need to learn the distribution of the three arrays and sample from the distributions separately. However, since the array $\{rotation_i\}$ are rotation ways of the array $\{carton_i\}$, sampling them separately will break the connection between them and lead to poor performance. We modified UMDA so that we sample new individuals by randomly picking genes from existing individuals which will not break the connection between array $\{rotation_i\}$ and $\{carton_i\}$. The detailed steps are as follows.

To sample the kth ordered variable in array $\{rotation_i\}$ and $\{carton_i\}$, a random number rnd between 0 and P is generated. If the variable in the kth position of array $\{carton_i\}$ in the rndth ordered individual has not appeared

in the sampled individual before, the sample individual's kth ordered variable in array $\{carton_i\}$ and $\{rotation_i\}$ equals to the rndth individual in existing population. If it had appeared before, we check the population in order, starting from individual rnd, until one individual with the kth ordered gene never appearing before is met or all individuals are checked. If all individuals are checked, a suitable random value in generated for position k for array $\{rotation_i\}$ and $\{carton_i\}$. The pseudo code for sampling array $\{rotation_i\}$ and $\{carton_i\}$ are as follows. The same steps are followed to sample array $\{bin\}$. After the three number sequences are sampled, we conduct mutation to them and then we finish sampling an individual.

The process of sampling array carton and array rotation

```
Sampling-individual( Pop[1..P]
    const
        P,N: Int;
    var
        i,j,temp,rnd: Int;
    begin
        i :=0;
        j :=0;
        repeat
            rnd := random(0,P);
            If pop[rnd].carton[j] never appeared
                individual.carton[j] := pop[rnd].carton[j];
                individual.rotation[j] := pop[rnd].rotation[j];
                return;
            Else temp :=rnd;
                repeat
                    temp :=temp+1;
                    if pop[temp].carton[j] never appeared
                        individual.carton[j] := population[temp].carton[j];
                        individual.rotation[j] := population[temp].rotation[j];
                        return
                until temp mod N ==rnd mod N
                assign carton[j] and rotation[j] randomly
            until j>=N
end.
```

For each generation, E best individuals are selected and added to the next population directly. $P - E$ individuals are sampled using the strategy mentioned above. After enough generations, a good enough solution can be achieved.

In the worst case, the time complexity of generating a gene is $O(P)$, thus the time complexity of of generating an individual is $O(m * P + N * P)$. In our algorithm, we generate P individuals for gen generations, so the time complexity of the algorithm is $O(gen * P * (m * P + N * P))$.

5 Numerical Examples and Analysis

In the following experimental study, we compared our algorithm with a GA approach and the CPLEX solver. The GA approach uses the same representation

strategy with our EDA and employs the crossover method mentioned in [6]. After the crossover is done, we conduct mutation operation to the number sequences. For every number in array $\{bin\}$ and $\{rotation\}$, it is switched position with another randomly selected number with probability pm. For every number in array $\{rotation\}$, it is changed to its another possible value with probability pm. Each population contains P individuals. Before generating the next population, E best individuals are directly added to the next population. In the experiment, the parameters for GA and modified UMDA are set as: $pm = 0.01$, $P = 100$, $E = 10$, $gen = 100$. The meaning of pm, P, E is in the above algorithm description. gen stands for generation which we repeat gen times of generating new population until we stop. The instances we use are from a transportation company. Each instance is tested for 10 times. The data shown in the Table 1 is the space utilization rate, which is calculated by $space_{items}/(space_{items} + space_{wasted})$. A larger value of the rate indicates a better quality of the solution. The time limit we set for CPLEX is 1800 seconds. If CPLEX failed to get a solution in 1800 seconds, we use a "-" to denote it.

Table 1. Comparison of space utilization percentage among three approaches

Instances	GA average	GA best	EDA average	EDA best	CPLEX	CPLEX gap
12-1	0.73	0.73	0.53	0.71	0.73	0
12-2	0.82	0.85	0.59	0.67	0.85	0
12-3	0.49	0.49	0.47	0.49	0.49	0
12-4	0.27	0.45	0.19	0.45	0.45	0
12-5	0.66	0.66	0.49	0.66	0.66	0
20-1	0.77	0.82	0.7	0.82	0.58	99%
20-2	0.76	0.77	0.67	0.77	0.62	99%
20-3	0.77	0.8	0.66	0.78	0.63	99%
20-4	0.68	0.72	0.6	0.77	0.7	98%
20-5	0.71	0.73	0.72	0.88	0.65	99%
50-1	0.69	0.78	0.74	0.83	-	-
50-2	0.7	0.76	0.74	0.83	-	-
50-3	0.68	0.73	0.75	0.85	-	-
50-4	0.72	0.76	0.73	0.8	-	-
50-5	0.62	0.7	0.66	0.75	-	-
78-1	0.65	0.68	0.7	0.74	-	-
78-2	0.66	0.69	0.71	0.74	-	-
78-3	0.65	0.68	0.74	0.82	-	-
78-4	0.67	0.7	0.7	0.74	-	-
78-5	0.67	0.72	0.72	0.79	-	-

From the experimental data, the following observations can be concluded: (1) For small instances with 12 items, CPLEX and GA are very likely to solve to optimal; modified UMDA can achieve optimal solution but its performance is not very stable. (2) For medium instances with 20 items, CPLEX can not find a good solution in 1800s. Both GA and modified UMDA can find a good solution. GA has a better "average solution" and modified UMDA has a better "best solution". (3) For large instances with 50 or 78 items, CPLEX failed to find a solution. Both GA and modified UMDA are able to find a solution. Modified UMDA has a better "average solution" and "best solution".

From the observations, it can be seen that the modified UMDA is a good choice for large instances. For small and medium instances, although the modified UMDA do not perform very stable, we can still get a solution better than or at least as good as GA and CPLEX by repeating the algorithm on the same instance for several times and pick out the best solution.

6 Conclusion

In this paper, bin packing problems with various bin sizes are investigated. The mathematical formulation and an estimation of distribution algorithm for the problem are proposed and experimentally studied. In the proposed algorithm, a packing method for multi bins is used for representation and a modified univariate marginal distribution algorithm is used for generating new population. The experiment results show that our proposed method seems to be a good optimizer for 3DBPP with various bin sizes, especially for a large instance with 50 or more items to pack.

References

1. Hifi, M., Kacem, I., Nègre, S., et al.: A linear programming approach for the three-dimensional bin-packing problem. Electronic Notes in Discrete Mathematics 36, 993–1000 (2010)
2. Martello, S., Pisinger, D., Vigo, D.: The three-dimensional bin packing problem. Operations Research 48(2), 256–267 (2000)
3. Lim, A., Rodrigues, B., Yang, Y.: 3-d container packing heuristics. Applied Intelligence 22(2), 125–134 (2005)
4. Lodi, A., Martello, S., Vigo, D.: Tspack: a unified tabu search code for multi-dimensional bin packing problems. Annals of Operations Research 131(1-4), 203–213 (2004)
5. Crainic, T., Perboli, G., Tadei, R.: TS2PACK: A two-level tabu search for the three-dimensional bin packing problem. European Journal of Operational Research 195(3), 744–760 (2009)
6. Wang, H.F., Chen, Y.J.: A hybrid genetic algorithm for 3d bin packing problems. In: 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), pp. 703–707. IEEE (2010)
7. Wu, Y., Li, W.K., Goh, M., Souza, R.: Three-dimensional bin packing problem with variable bin height. European Journal of Operational Research 202(2), 347–355 (2010)
8. Henrion, M.: Propagating uncertainty in bayesian networks by probabilistic logic sampling. In: Uncertainty in Artificial Intelligence, vol. 2, pp. 149–163 (1988)
9. Mühlenbein, H.: The equation for response to selection and its use for prediction. Evolutionary Computation 5(3), 303–346 (1997)

Accelerating BIRCH for Clustering Large Scale Streaming Data Using CUDA Dynamic Parallelism

Jianqiang Dong, Fei Wang, and Bo Yuan

Intelligent Computing Lab, Division of Informatics
Graduate School at Shenzhen, Tsinghua University
Shenzhen 518055, P.R. China
513712287@qq.com, wangfeifast@gmail.com,
yuanb@sz.tsinghua.edu.cn

Abstract. In this big data era, the capability of mining and analyzing large scale datasets is imperative. As data are becoming more abundant than ever before, data driven methods are playing a critical role in areas such as decision support and business intelligence. In this paper, we demonstrate how state-of-the-art GPUs and the Dynamic Parallelism feature of the latest CUDA platform can bring significant benefits to BIRCH, one of the most well-known clustering techniques for streaming data. Experiment results show that, on a number of benchmark problems, the GPU accelerated BIRCH can be made up to 154 times faster than the CPU version with good scalability and high accuracy. Our work suggests that massively parallel GPU computing is a promising and effective solution to the challenges of big data.

Keywords: GPU, CUDA, Dynamic Parallelism, BIRCH, Big Data, Clustering.

1 Introduction

In the era of big data, modern organizations in almost all industries are facing increasingly growing amount of heterogeneous data at unprecedented speed. Each day, around 2.5 quintillion bytes of data[1] are created such as sensor data, posts in social networks, digital images/videos, web search results, telecommunication records and financial transactions. The scale of the data available creates significant challenges for traditional techniques to effectively store, transfer, visualize and analyze the data within a reasonable amount of time.

Despite of the large number of existing data mining algorithms for tasks such as classification, clustering and frequent pattern analysis, there are two major issues that must be carefully addressed before they can be properly applied in the scenario of big data. Firstly, many algorithms assume that all data are stored in the main memory, which can be readily accessed. However, for real-world problems, the size of the data can easily exceed the memory capacity and, when multiple access to the dataset is required, the I/O cost may severely compromise the efficiency of the algorithm. To solve this issue, various data stream mining techniques have been proposed, which

[1] http://www-01.ibm.com/software/data/bigdata/

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 409–416, 2013.

only require reading the data once. They are particularly suitable for situations where the entire dataset is too large to fit into the main memory or the data come in a continuous manner.

Secondly, most data mining algorithms are designed without explicitly taking parallel computing into account, although they may have inherent potential for parallelism. It is often assumed that each computing step is to be executed sequentially and no special efforts are devoted to harnessing the power of advanced multi-core and many-core computing devices that are becoming increasingly popular in the past decade. As a result, even with seemingly decent computational complexity in theory, the real running time of these algorithms on non-trivial datasets can be prohibitively intolerable. In fact, this issue is creating a large gap between academic research in data mining and industrial applications.

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [1, 2] is one of the most well-known hierarchical clustering algorithms for large scale data, which can incrementally cluster incoming data and requires only a single scan of the dataset in most cases. To make BIRCH more applicable on real-world problems, in this paper, we will investigate how to effectively accelerate BIRCH using parallel computing techniques.

In addition to CPU-based parallel computing architecture such as MPI[2] and OpenMP[3], in recent years, GPU (Graphics Processing Unit) computing is quickly becoming a new powerhouse for providing high performance computing capability at dramatically reduced cost and many GPU-based data mining algorithms have been proposed [3, 4]. Modern GPUs feature thousands of cores and support tens of thousands concurrent threads (many-core computing), making them specifically suitable for massively data parallel computing tasks. In many application areas such as fluid dynamics, financial engineering, life science and signal processing, researchers can often obtain 10~100× speedups on computing intensive problems by using standard workstations equipped with advanced GPU computing cards. In nowadays, the peak performance of high-end GPUs is over one TFLOPS (double-precision floating point) and the average cost to achieve one GFLOPS is already less than one dollar with GPU computing[4].

In the rest part of this paper, Section 2 gives a brief review on clustering algorithms for large scale data, especially the BIRCH algorithm. Section 3 introduces CUDA[5] (Compute Unified Device Architecture) and the Dynamic Parallelism technique, which bring incredible convenience to GPU computing with measurable performance improvement. The parallel implementation of BIRCH is detailed in Section 4 along with the experiment specification. The main experiment results are presented in Section 5 and this paper is concluded in Section 6 with some discussions and directions for future work.

[2] http://www.mcs.anl.gov/research/projects/mpi/
[3] http://www.openmp.org/
[4] http://en.wikipedia.org/wiki/FLOPS#Cost_of_computing/
[5] https://developer.nvidia.com/cuda-toolkit/

2 Clustering Big Data

Clustering is one of the most important unsupervised learning methods in pattern recognition and data mining [5]. There are mainly two categories of clustering methods: hierarchical clustering such as agglomerative clustering and partition-based clustering such as K-means. For data stream mining [6-8], CluStream [9] is an important hierarchical clustering algorithm, which uses many micro clusters to form a better macro cluster. In the meantime, the extended K-Means algorithm and STREAM [10] are two good examples of partition-based methods.

In BIRCH, a key concept is called clustering feature (CF), which is a triple holding the necessary information (e.g., linear sum and square sum) of all data points belonging to a certain cluster. The main procedure of BIRCH involves constructing a height balanced tree called CF-Tree. A parent node in the CF-Tree stores the summary of a large cluster while each child node maintains the information of its own small cluster. There are two major parameters in BIRCH named *B* and *T*. *B* is the branching factor of the CF-Tree, indicating the maximum number of children of the parent node. *T* is the threshold used to determine whether a new coming data point can be absorbed by an existing cluster.

In recent years, a few GPU implementations of hierarchical clustering algorithms have been proposed. For example, a speedup of more than 100 times was achieved on the agglomerative clustering algorithm [11, 12]. In the meantime, MPI and the Thrust library[6] have been used to accelerate BIRCH on various tasks such as text clustering problems [13, 14]. However, as Thrust can only parallelize certain components in BIRCH and it takes extra time to transfer data between host (CPU) and device (GPU), the overall speedup was only around 6 times.

3 GPU Computing

CUDA is a GPU programming environment developed by NVIDIA in 2007 and is the most widely used platform for GPU programming. The latest CUDA version is 5.5, which is the best match to GPUs with compute capability 3.5 such as Tesla K20 and GeForce GTX TITIAN. Please refer to CUDA C Computing Guide[7] and CUDA C Best Practices Guide[8] for a comprehensive review.

With the increasing popularity of CUDA, more and more computing libraries have been developed, such as Thrust, cuRAND[9] and so on. Thrust is a CUDA STL programming model, which can handle the data on GPU in a C++ STL way, and consists of many commonly used algorithms such as sorting and reduction. cuRAND is a CUDA random number library, which can generate various kinds of random number in a very efficient way.

---

[6] `https://developer.nvidia.com/thrust/`
[7] `http://docs.nvidia.com/cuda/cuda-c-programming-guide/`
[8] `http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/`
[9] `https://developer.nvidia.com/curand`

CUDA Dynamic Parallelism[10] and Kepler Compute Architecture[11] together create the state-of-the-art GPU computing environment, making GPU computing much more efficient and easier to code compared to the last generation GPU architecture. Dynamic Parallelism offers exciting new capabilities by providing a mechanism for calling kernel functions from another kernel function so that GPU programs can be made more flexible and easier to synchronize. Fig. 1 shows an illustration of kernel launching in GPU.

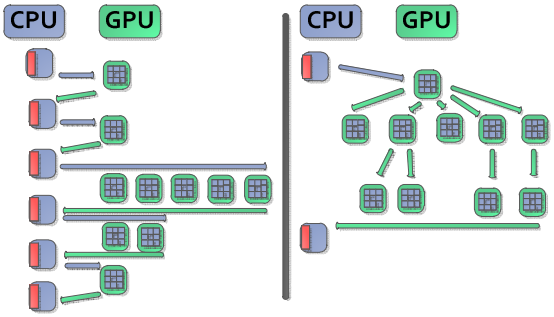


Fig. 1. Kernel launching with (right) and without dynamic parallelism (left)

4 Methodology

4.1 Standard BIRCH

The main operations in BIRCH include the construction of the CF Tree and assigning data points to a nearby CF value. In Table 1, each data point has to go through step 3 and step 4. Most data points are allocated into an existing cluster (step 5.1) while some data points create a new CF (step 5.2), and only very few data points result in the splitting of the CF tree and creating a new CF (step 5.3).

Table 1. The framework of BIRCH

Step	*BIRCH*
1	Initialize the data source.
2	For each data sample.
3	While (not leaf)
3.1	Find the nearest CF.
3.2	Go down to the nearest CF sub-tree.
4	Find the nearest CF in the leaf node.
5	Compute the distance *t* between the data point and the nearest CF
5.1	If *t*<=T then absorb.
5.2	If *t*>T and Leaf node is not full, then add new CF.
5.3	Else split the leaf node to allocate the new nodes.

[10] `http://docs.nvidia.com/cuda/cuda-dynamic-parallelism/index.html`

[11] `http://www.nvidia.com/object/nvidia-kepler.html`

4.2 GPU Accelerated BIRCH

In the proposed GPU based BIRCH (GBIRCH), a master kernel is launched first. Next, several slave kernels are launched using CUDA Dynamic Parallelism, with each slave kernel dealing with a subset of the data samples in the GPU memory. The slave kernel is terminated after its associated data samples have all been processed. Tables 2-4 present the pseudo code of the master kernel, the slave kernel, and the refinery kernel.

Table 2. Master kernel

Step	*GBIRCH_Master*
1	Set up the data source and partition the data.
2	Build up the CF Tree.
3	For each data block
3.1	Call *GBIRCH_Slave*.
4	Call *GBIRCH_Refinary*.
5	Output the data point with its cluster information.
6	Repeat steps 3~5, until all the data are processed.

Slave kernels are responsible for most of the computing tasks in BIRCH. Each slave kernel fetches some data records from the master kernel and computes the distances and finds the nearest CF Leaf. If the data record can be absorbed by an existing cluster, the slave kernel does the job individually. If a new CF is to be added to the CF tree, or the CF tree needs to be split, the data will be transferred back to the master kernel. In this way, the parallel slave kernels can always keep working on the same and well organized CF tree.

Table 3. Slave kernel

Step	*GBIRCH_Slave*	*Memory Allocation*
1	Get data from *GBIRCH_Master*.	Global to Shared
2	For each data sample	
3	While (not leaf)	Global to Shared
3.1	Find the nearest CF.	Shared and Register
3.2	Go down to the nearest CF sub-tree.	Global to Shared
4	Find the nearest CF in the leaf node.	Shared and Register
5	Compute t.	On a single thread
5.1	If t<=T, absorb the data sample.	
5.2	If t>T and Leaf is not full, add a new CF.	
5.3	Else save data point and submit to *GBIRCH_Master*.	

The refinery kernel keeps the CF tree accurate and well organized. Step 1 happens more frequently than other steps and is parallelized. Steps 2~4 are used to split the CF tree and no two threads can work on the CF tree at the same time (locking is required to ensure mutual exclusion).

Table 4. Refinery kernel

Step	*GBIRCH_Refinery*
1	Test the new CF Leaf from *GBIRCH_Slave*. If the CF parent is full, then add to split candidate.
2	Each thread begins to split with a candidate, and lock the CF tree when it is used in this process.
3	Roll up to split the higher level nodes, until reaching Root.
4	Repeat steps 2~3, until all the candidates are split.

In the CF tree, each node uses an array to store the CF values of its children and keeps *parentID*, *childID* and the number of children. In our work, the branching value was between 16~128. A smaller value may severely reduce the potential for parallelism while a larger value may result in a CF tree that is not deep enough to produce fine clustering granularity.

4.3 Benchmark Datasets

In the experiments, we used both synthetic and real datasets and all six datasets are widely used in data mining research. The first three datasets (DS1, DS2, DS3) were used in the original work on BIRCH [1, 2]. Each dataset consists of many centroids (from 100 to 1 million), and each centroid has 100 to 1 million data points. KDD CUP is a well-recognized annual international knowledge discovery and data mining competition. We used KDD CUP 98 data[12] (191,781 samples, 481 attributes), KDD CUP 99 dataset[13] (4,898,431 samples, 42 attributes) and KDD CUP 2012 data[14] (149,639,105 samples, 12 attributes) in our experiments.

5 Experiment Results

The software environment was: CUDA 5.0, Visual Studio 2010 and Windows 7. The hardware configuration was: Intel Core i5-2320 (CPU), 8GB RAM and NVIDIA Tesla K20 (Kepler Architecture GPU).

On the six datasets, the maximum speedup value of GBIRCH was 154.29 on KDD CUP 2012 data set (Table 5). In general, the results were better on larger datasets as the CF tree construction process is difficult to be executed in parallel on GPU (as the dataset gets larger, more time is spent on the absorbing process rather than the construction of the CF tree).

We also tested the scalability of GBIRCH using DS3. In Table 6, it is clear that as the number of records increases (the dimension was fixed to 2), the advantage of GBIRCH over BIRCH became more evident and as the dimension increased, the relative performance of GBIRCH was kept reasonably stable. Finally, the accuracy of GBIRCH was evaluated by measuring the percentage of data points that were

[12] `http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html`
[13] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
[14] `http://www.kddcup2012.org/c/kddcup2012-track2/data`

assigned to the same cluster as in the CPU version. Table 7 shows that the impact of parallelization on the accuracy of BIRCH is trivial.

Table 5. Comparison of running time (ms)

Dataset	*CPU*	*GPU*	*Speedup*
DS1	498691	3983	125.20
DS2	502145	4486	111.93
DS3	501168	4296	116.66
KDD CUP 98	21318	2796	7.62
KDD CUP 99	632264	8512	74.28
KDD CUP 2012	3268765	21186	154.29

Table 6. Scalability test with regard to the number of records and dimensionality (ms)

# of Data	*CPU*	*GPU*	*Speedup*	*Dim*	*CPU*	*GPU*	*Speedup*
1,000,000	12401	1602	7.74	2	12401	1602	7.74
4,000,000	49820	1692	19.44	5	19185	2448	7.84
12,000,000	150543	2088	72.10	10	28129	3918	7.18
40,000,000	501168	4296	116.66	25	55224	8202	6.73

Table 7. Accuracy of GBIRCH

Dataset	*Total Number*	*Correct Number*	*Accuracy*
KDD CUP 98	191,781	191,781	100%
KDD CUP 99	4,898,431	4,897,012	99.9%
KDD CUP 2012	149,639,105	148,527,097	99.3%

6 Conclusions

The major motivation of this paper was to investigate an important question in the era of big data: how to effectively handle the challenges from clustering large scale data. In our work, we chose BIRCH, a well-known clustering technique for streaming data, as an example to show how advanced GPUs and CUDA platform with the latest Dynamic Parallelism capability can significantly reduce the time required for clustering large datasets. Experiment results demonstrated that, with a careful parallel implementation of the major procedures in BIRCH and the help of Dynamic Parallelism as well as the smart use of GPU memory, GBIRCH achieved encouraging speedups from 7 to 154 times over the original BIRCH on six benchmark datasets. In the meantime, GBIRCH also featured satisfactory scalability with regard to the size and dimensionality of the dataset.

There are a number of possible directions for future work. For example, we can implement the GPU versions of other popular data stream clustering algorithms such as CluStream or CURE [15]. The performance of GPU accelerated clustering algorithms on other data types such as text and XML or on high dimensional datasets is also worth investigation.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 60905030) and NVIDIA CUDA Teaching Center Program.

References

1. Zhang, T., Raghu, R., Miron, L.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. ACM SIGMOD Record 25(2), 103–114 (1996)
2. Zhang, T., Raghu, R., Miron, L.: BIRCH: A New Data Clustering Algorithm and Its Applications. Data Mining and Knowledge Discovery 1(2), 141–182 (1997)
3. Fang, W., Lau, K., Lu, M., et al.: Parallel Data Mining on Graphics Processors. Technical Report HKUST-CS08-07 (2008)
4. Bai, H., He, L., Ouyang, D., Li, Z., Li, H.: K-Means on Commodity GPUs with CUDA. In: 2009 WRI World Congress on Computer Science and Information Engineering, pp. 651–655 (2009)
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
6. Mahdiraji, A.R.: Clustering Data Stream: A Survey of Algorithms. International Journal of Knowledge-Based and Intelligent Engineering Systems 13(2), 39–44 (2009)
7. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Kogan, J., et al. (eds.) Grouping Multidimensional Data, pp. 25–71. Springer (2006)
8. Barbará, D.: Requirements for Clustering Data Streams. ACM SIGKDD Explorations Newsletter 3(2), 23–27 (2002)
9. Aggarwal, C.C., Han, J., Wang, J., Yu, P.: A Framework for Clustering Evolving Data Streams. In: 29th International Conference on Very Large Data Bases, pp. 81–92 (2003)
10. O'Callaghan, L., Meyerson, A., Motwani, R., Mishra, N., Guha, S.: Streaming-Data Algorithms for High-Quality Clustering. In: 18th International Conference on Data Engineering, pp. 685–694 (2002)
11. Shalom, S.A., Dash, M.: Efficient Partitioning Based Hierarchical Agglomerative Clustering Using Graphics Accelerations with CUDA. International Journal of Artificial Intelligence & Applications 4(2), 13–33 (2013)
12. Shalom, S.A., Dash, M., Tue, M., Wilson, N.: Hierarchical Agglomerative Clustering Using Graphics Processor with Compute Unified Device Architecture. In: 2009 International Conference on Signal Processing Systems, pp. 556–561 (2009)
13. Garg, A., Mangla, A., Gupta, N., Bhatnagar, V.: PBIRCH: A Scalable Parallel Clustering Algorithm for Incremental Data. In: 10th IEEE International Database Engineering and Applications Symposium, pp. 315–316 (2006)
14. Bagga, A., Toshniwal, D.: Parallelization of Hierarchical Text Clustering on Multi-core CUDA Architecture. International Journal of Computer Science and Electrical Engineering 1, 72–76 (2012)
15. Guha, S., Rastogi, R., Shim, K.: CURE: An Efficient Clustering Algorithm for Large Databases. In: 1998 ACM International Conference on Management of Data, pp. 73–84 (1998)

Swarm Intelligence in Big Data Analytics

Shi Cheng[1,⋆], Yuhui Shi[2], Quande Qin[3], and Ruibin Bai[1]

[1] Division of Computer Science, University of Nottingham Ningbo, China
[2] Department of Electrical & Electronic Engineering,
Xi'an Jiaotong-Liverpool University, Suzhou, China
[3] College of Management, Shenzhen University, Shenzhen, China
shi.cheng@nottingham.edu.cn, yuhui.shi@xjtlu.edu.cn

Abstract. This paper analyses the difficulty of big data analytics problems and the potential of swarm intelligence solving big data analytics problems. Nowadays, the big data analytics has attracted more and more attentions, which is required to manage immense amounts of data quickly. However, current researches mainly focus on the amount of data. In this paper, the other three properties of big data analytics, which include the high dimensionality of data, the dynamical change of data, and the multi-objective of problems, are discussed. Swarm intelligence, which works with a population of individuals, is a collection of nature-inspired searching techniques. It has effectively solved many large-scale, dynamical, and multi-objective problems. Based on the combination of swarm intelligence and data mining techniques, we can have better understanding of the big data analytics problems, and designing more effective algorithms to solve real-world big data analytics problems.

1 Introduction

Nowadays, the big data analytics has attracted attentions from more and more researchers. The big data is defined as the dataset whose size is beyond the processing ability of typical database or computers. Four objects are emphasized in the definition, which are capture, store, management, and analysis [26]. However, the definition only focuses on the amount of data. There are three other properties which also need to be emphasized in the big data analytics research.

The dimensionality of data affects the performance of algorithms. Many methods suffer from the "curse of dimensionality", which implies that their performance deteriorates quickly as the dimension of the search space increases [12,18]. The big data analytics also suffers this problem. Handling large scale data with a good performance in limited time should be concerned in the big data analytics.

The content of the big data is increasing over time, and the target of big data analytics also changes with time. The algorithm should be able to handle the dynamical changing data, and to adjust the target of data analytic.

⋆ The authors' work is partially supported by National Natural Science Foundation of China under Grant Number 60975080, 61273367; and by Ningbo Science & Technology Bureau (Science and Technology Project Number 2012B10055).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 417–426, 2013.

The data came from different sources in the large dataset. Generally, more than one objective needs to be satisfied at the same time in these large dataset. The most traditional methods can only be applied to continuous and differentiable functions, and have to perform a series of separate runs to satisfy different objectives.

Swarm intelligence is a set of search and optimization techniques [14,15,22]. To search a problem domain, a swarm intelligence algorithm processes a *population* of individuals. Different from traditional single-point based algorithms such as hill-climbing algorithms, each swarm intelligence algorithm is a population-based algorithm, which consists of a set of points (population of individuals). Each individual represents a potential solution of the problem being optimized. The population of individuals is expected to have high tendency to move towards better and better solution areas iteration over iteration through cooperation and/or competition among themselves.

In this paper, the difficulty of big data analytics problem is analysed. Big data analytics are divided into four components: handling large amount of data, handling high dimensional data, handling dynamical data, and multi-objective optimization. Most real world big data problems can be modelled as a large scale, dynamical, and multi-objective problems. Swarm intelligence has shown significant achievement on these problems. With the swarm intelligence, more effective methods can be designed and utilized in the big data analytical problems.

This paper is organized as follows. Section 2 and 3 review the basic concepts of big data analytics and the swarm intelligence methods, respectively. Swarm intelligence utilized in the big data analytics problems is introduced in Section 4. Two applications of big data analytics, the intelligent transport system and the wireless sensors networks, are brief reviewed in Section 5, followed by conclusions in Section 6.

2 Big Data Analytics

The big data is defined as the dataset whose size is beyond the processing ability of typical database or computers. Four objects are emphasized in the definition, which are capture, store, management, and analysis [26]. The big data analytics is to automatically extract knowledge from large amounts of data. It can be seen as mining or processing of massive data, and "useful" information could be retrieved from large dataset [29]. The properties of big data analytics can be concentrated in three parts: large volume, variety of different sources, and fast increasing speed, i.e., velocity. The algorithms should be effective to solve large-scale, dynamic big data analytics problems.

The knowledge discovery in databases (KDD) is the process of converting raw data into useful information. Data mining (the analysis step of KDD), is the process that attempts to discover useful information (or patterns) in large data repositories [16]. The data mining field includes many subfields, such as classification analysis, clustering analysis, and association analysis, just to name a few.

The big data may contain many kinds of unstructured or semi-structured data; these data need to be transformed as structured data. A kind of data's attribute will be transformed as a feature of data, and thus an example of data is transformed as a vector which contains many features. The dimension of the feature space is equal to the number of different attributes that can appear in the data set. This indicates that the dimension of big data analytics problems is much higher than the traditional problems.

The data clustering methods also can be applied to the swarm intelligence [32]. In the brain storm optimization algorithm, every solution is spread in the search space. The distribution of solutions can be utilized to reveal the landscapes of a problem. From the clustering analysis of solutions, the search results can be obtained [7].

3 Swarm Intelligence

Many real-world applications can be represented as optimization problems of which algorithms are required to have the capability to search for optimum. Most traditional methods can only be applied to continuous and differentiable functions [32]. The meta-heuristic algorithms are proposed to solve the problems, which the traditional methods cannot solve or at least be difficult to solve. Recently, kind of meta-heuristic algorithms, termed as swarm intelligence, are attracting more and more attentions from researchers.

Swarm intelligence (SI), which is based on a population of individuals, is a collection of nature-inspired searching techniques [22]. To search a problem domain, a swarm intelligence algorithm processes a *population* of individuals. Each individual represents a potential solution of the problem being optimized. In swarm intelligence, an algorithm maintains and successively improves a population of potential solutions until some stopping condition is met. The solutions are initialized randomly in the search space, and are guided toward the better and better areas through the interaction among solutions over iterations [5, 32].

As a general principle, the expected fitness of a solution returned should improve as the search method is given more computational resources in time and/or space. More desirable, in any single run, the quality of the solution returned by the method over iterations should improve monotonically – that is, the quality of the solution at time $t + 1$ should be no worse than the quality at time t, i.e., $fitness(t + 1) \leq fitness(t)$ for minimum problems [17]. There exist many swarm intelligence algorithms, among them most common ones are the particle swarm optimization (PSO) algorithm [21], which was originally designed for solving continuous optimization problems, and the ant colony optimization (ACO) algorithm, which was originally designed for discrete optimization problems [13].

The most important factor affecting a swarm intelligence algorithm's performance may be its ability of exploration and exploitation [6]. Exploration means the ability of a search algorithm to explore different areas of the search space in order to have high probability to find good promising solutions. Exploitation, on the other hand, means the ability to concentrate the search around a promising

region in order to refine a candidate solution. A good optimization algorithm should optimally balance the two conflicted objectives.

4 Swarm Intelligence in Big Data Analytics

Data mining has been a popular academic topic in computer science and statistics for decades, swarm intelligence is a relatively new subfield of computational intelligence (CI) which studies the collective intelligence in a group of simple individuals. In the swarm intelligence, useful information can be obtained from the competition and cooperation of individuals.

Generally, there are two kinds of approaches that apply swarm intelligence as data mining techniques [27]. The first category consists of techniques where individuals of a swarm move through a solution space and search for solution(s) for the data mining task. This is a search approach; the swarm intelligence is applied to optimize the data mining technique, e.g., the parameter tuning. In the second category, swarms move data instances that are placed on a low-dimensional feature space in order to come to a suitable clustering or low-dimensional mapping solution of the data. This is a data organizing approach; the swarm intelligence is directly applied to the data samples, e.g., dimensionality reduction of the data.

Swarm intelligence, especially particle swarm optimization or ant colony optimization algorithms, is utilized in data mining to solve single objective [1] and multi-objective problems [9]. Based on the two characters of particle swarm, the self-cognitive and social learning, the particle swarm has been utilized in data clustering techniques [10], document clustering, variable weighting in clustering high-dimensional data [25], semi-supervised learning based text categorization, and the Web data mining [28].

In a swarm intelligence algorithm, there are several solutions exist at the same time. The premature convergence may happen due to the solution getting clustered together too fast. However, the solution clustering is not always harmful for optimization. In a brain storm optimization algorithm, the clustering analysis is utilized to reveal the landscapes of problems and to guide the individuals to move toward the better and better areas [32]. Every individual in the brain storm optimization algorithm is not only a solution to the problem to be optimized, but also a data point to reveal the landscapes of the problem [7]. The machine learning and data mining techniques can be combined to produce benefits above and/or beyond what either method could achieve alone [31].

The big data analytics is required to manage immense amounts of data quickly [29]. The amount of data are attracting more and more attentions, however, the dimension of data and the number of objective of problems also increase the "hardness" of problems. Three kinds of difficulties should be overwhelmed to solve big data problems:

4.1 Large Scale Optimization

The big data analytics requires a fast mining on the large scale dataset, i.e., the immense amounts of data should be processed in a limited time. The analytic

problem can be modelled as optimization problems. In general, optimization concerns with finding the "best available" solution(s) for a given problem within allowable time, and the problem may have several or numerous optimum solutions, of which many are local optimal solutions. Normally, the difficulty of problem will increase with the increasing of the number of variables and objectives. Specially, problems with large number of variables, e.g., more than thousands variables, are termed as large scale problems.

Many optimization methods suffer from the "curse of dimensionality", which implies that their performance deteriorates quickly as the dimension of the search space increases [2,11,18]. There are several reasons that cause this phenomenon.

First, the solution space of a problem often increases exponentially with the problem dimension and more efficient search strategies are required to explore all promising regions within a given time budget. The evolutionary computation or swarm intelligence is based on the interaction of a group of solutions. The promising regions or the landscape of problems are very difficult to reveal by small solution samples (compared with the number of all feasible solutions).

The "empty space phenomenon" gives an example of problems getting hard when the dimension increases [30]. The number of possible solutions is increased exponentially when the dimension increasing. The search performance of most algorithms is based on the previous search experience. Considered the limitation of computational resources, the percentages of data points have been retrieved will close to zero when the dimension increased to a large number. The performance of algorithms is affected by the increasing of problems' dimension.

Second, the characteristics of a problem may change with the scale. Problems will become more difficult and complex when the dimension increases. Rosenbrock's function, for instance, is unimodal for two dimensional problems but becomes multimodal for higher dimensional problems. Because of such a worsening of the features of an optimization problem resulting from an increase in scale, a previously successful search strategy may no longer be capable of finding an optimal solution.

Third, the direction of "good" solutions is difficult to determine. The swarm intelligence takes an "update on each dimension, evaluation on whole dimensions" strategy. An algorithm is very difficult to determine which one is better when two solutions both have some good parts and their fitness values are equally bad. The similar scenario also happens in multiobjective optimization. In Pareto domination measurement, nearly all solutions are Pareto non-dominated when the number of objects is larger than 10.

The last, the bias is accumulated. In the swarm intelligence, each solution is updated dimension by dimension, and the fitness value is calculated for the whole solution. The solution update depends on the combination of several vectors, i.e., the current value, the difference between current value and previous best value, the differential between current value and neighbor best value, or the difference between two random solutions, etc. In the low dimensional space, the direction of the vector combination has the high probability to point to the global optimum. However, the distance metric, which is utilized in low dimension space, is not

effective in high dimensional space. The search direction is far away from the global optimum due to the bias accumulation.

Many effective strategies are proposed for high dimensional optimization problems, such as problem decomposition and subcomponents cooperation, parameter adaptation, surrogate-based fitness evaluations [20]. Based on the swarm intelligence, an effective method could find good solutions for large scale problems, both on the time complexity and result accuracy.

4.2 Handling High Dimensional Data

The "curse of dimensionality" also happens on the high dimensional data mining problems [11, 12, 18]. Many algorithms' performance deteriorates quickly as the dimension of the data space increases. For example, the nearest neighbor approaches are very effective in categorization. However, for high dimensional data, it is very difficult to solve the similarity search problem due to the computational complexity, which was caused by the increase of dimensionality.

Many methods are proposed on the high dimension data mining problems. Transforming the high dimensional mining problems into low dimensional space via a "projection" operation is an effective way. The locality sensitive hashing algorithm is proposed to find nearest neighbors in the high dimensional space [33]. This algorithm is based on hashing functions with strong "local-sensitivity" in order to retrieve nearest neighbors in a Euclidean space with a complexity sublinear in the amount of data.

The data mining problem can be transformed as an optimization problem, because many researches have been taken on the large scale optimization problems. Swarm intelligence, especially particle swarm optimization or ant colony optimization algorithms, is utilized in data mining to solve single objective [1] and multiobjective problems [9].

4.3 Handling Dynamical Data

The big data, such as the web usage data of Internet, real time traffic information, rapidly changes over time. The analytical algorithms need to process these data swiftly. The dynamic problems, sometimes termed as non-stationary environments, or uncertain environments [19], dynamically change over time. Swarm intelligence has been widely applied to solve stationary and dynamical optimization problems.

Swarm intelligence often has to solve optimization problems in the presence of a wide range of uncertainties. Generally, uncertainties in optimized problems can be divided into the following categories.

1. The fitness function or the processed data is noisy.
2. The design variables and/or the environmental parameters may change after optimization, and the quality of the obtained optimal solution should be robust against environmental changes or deviations from the optimal point.

3. The fitness function is approximated [20], such as surrogate-based fitness evaluations, which means that the fitness function suffers from approximation errors.
4. The optimum in the problem space may change over time. The algorithm should be able to track the optimum continuously.
5. The target of optimization may change over time. The demand of optimization may adjust to the dynamical environment, for example, there should be a balance between the computing efficiency and the computational cost for different computing loads.

In all these cases, additional measures must be taken so that swarm intelligence algorithms are still able to solve satisfactorily dynamic problems [19].

4.4 Multi-objective Optimization

Different sources of data are integrated in the big data research, and in most of the big data analytics problems, more than one objective need to be satisfied at the same time. According to the number of objectives, optimization problems can be divided as single objective and multiobjective problems. For the multiobjective problems, the traditional mathematical programming techniques have to perform a series of separate runs to satisfy different objectives.

Multiobjective Optimization refers to optimization problems that involve two or more objectives, and a set of solutions is sought instead of one. A general *multiobjective optimization problem* can be described as a vector function $\mathbf{f}$ that maps a tuple of n parameters (decision variables) to a tuple of k objectives.

Unlike the single objective optimization, the multiobjective problems have many or infinite solutions [3]. The optimization goal of an MOP consists of three objectives:

1. The distance of the resulting nondominated solutions to the true optimal Pareto front should be minimized;
2. A good (in most cases uniform) distribution of the obtained solutions is desirable;
3. The spread of the obtained nondominated solutions should be maximized, i.e., for each objective a wide range of values should be covered by the nondominated solutions.

In a multiobjective optimization problem, we aim to find the set of optimal tradeoff solutions known as the Pareto optimal set. Pareto optimality is defined with respect to the concept of nondominated points in the objective space. Swarm intelligence methods can effectively solve the multiobjective problems. Several new techniques are combined in the swarm intelligence techniques to solve multiobjective problems with more than ten objectives, in which almost every solution is Pareto nondominated in the problems. These techniques include objective decomposition, objective reduction [4], and clustering in the objective space [32].

5 Applications

The big data is created in many fields in everyday life. With the big data analytics techniques and swarm intelligence methods, more effective applications or systems can be designed to solve real world problems. The intelligent transportation system and wireless sensor networks are two typical examples of big data analytics application.

5.1 Intelligent Transportation System

The traffic problems are arising in many cities now. The traffic and transportation system is affected by many factors, such as the number of vehicles, weather, accidents, etc., and the traffic information changes in real time. The purpose of intelligent transport is to build more rapid, safe, and more efficient traffic and transportation systems by constructing the intelligent vehicles and road environment [34]. There are more than one objectives which need to be satisfied at the same time in intelligent transport systems, for example, rapid transportation, environmental pollution, transportation scheduling; and many of these objectives are conflicted with each other.

5.2 Wireless Sensor Networks

Based on the wireless sensor networks, the physical world is turning to be a kind of information system [8]. Different sensors are connected to form a networks; information is transferred in this network by communication techniques. The physical world's information from sensor networks can be collected almost anywhere at any time. The sensor networks and communication techniques have constructed a new paradigm, which is called the internet of things [8].

The wireless sensor networks have been applied to many real-world problems, such as environmental surveillance, transportation monitoring, engineering surveying, and industrial control, just to name a few [24]. Massive data will be generated from the long term and/or large scale wireless sensor network system. The goal of data analysis is to make the fastest possible revelation toward the "useful" information. Swarm intelligence is an effective way to handle these data, and to obtain "useful" information [23].

6 Conclusions

The big data has attracted more and more attentions currently. Most of the big data researches focus on the huge amount of data, however, handling the high dimensional data and the multiple objectives are also important in solving big data problems.

In this paper, the difficulty of big data analytics problem is analysed. Big data analytics are divided into four components: handling large amount of data, handling high dimensional data, handling dynamical data, and multi-objective

optimization. Most real world big data problems can be modelled as a large scale, dynamical, and multi-objective problems.

This paper is not to survey what have been done in the past, but to suggest the potential of swarm intelligence in big data analytics. Big data involves high-dimensional problems and a large amount of data. Swarm intelligence studies the collective behaviours in a group of individuals. It has shown significant achievements on solving large scale, dynamical, and multi-objective problems. With the application of the swarm intelligence, more rapid and effective methods can be designed to solve big data analytics problems.

References

1. Abraham, A., Grosan, C., Ramos, V. (eds.): Swarm Intelligence in Data Mining. SCI, vol. 34. Springer, Heidelberg (2006)
2. Bellman, R.: Adaptive Control Processes: A guided Tour. Princeton University Press, Princeton (1961)
3. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(2), 174–188 (2003)
4. Brockhoff, D., Zitzler, E.: Objective reduction in evolutionary multiobjective optimization: Theory and applications. Evolutionary Computation 17(2), 135–166 (2009)
5. Cheng, S.: Population Diversity in Particle Swarm Optimization: Definition, Observation, Control, and Application. Ph.D. thesis, Department of Electrical Engineering and Electronics, University of Liverpool (May 2013)
6. Cheng, S., Shi, Y., Qin, Q.: Population diversity of particle swarm optimizer solving single and multi-objective problems. International Journal of Swarm Intelligence Research (IJSIR) 3(4), 23–60 (2012)
7. Cheng, S., Shi, Y., Qin, Q., Gao, S.: Solution clustering analysis in brain storm optimization algorithm. In: Proceedings of The 2013 IEEE Symposium on Swarm Intelligence (SIS 2013), pp. 111–118. IEEE, Singapore (2013)
8. Chui, M., Löffler, M., Roberts, R.: The internet of things. McKinsey Quarterly 2, 1–9 (2010)
9. Coello, C.A.C., Dehuri, S., Ghosh, S. (eds.): Swarm Intelligence for Multi-objective Problems in Data Mining. SCI, vol. 242. Springer, Heidelberg (2009)
10. Cohen, S.C.M., de Castro, L.N.: Data clustering with particle swarms. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computations (CEC 2006), pp. 1792–1798 (July 2006)
11. Domingos, P.: A few useful things to know about machine learning. Communications of the ACM 55(10), 78–87 (2012)
12. Donoho, D.L.: Aide-Memoire. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Tech. rep., Stanford University (August 2000)
13. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(1), 29–41 (1996)
14. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press (June 2004)
15. Eberhart, R., Shi, Y.: Computational Intelligence: Concepts to Implementations, 1st edn. Morgan Kaufmann Publisher (2007)

16. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Magazine 17(3), 37–54 (1996)
17. Ficici, S.G.: Monotonic solution concepts in coevolution. In: Genetic and Evolutionary Computation Conference (GECCO 2005), pp. 499–506 (June 2005)
18. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer Series in Statistics. Springer (February 2009)
19. Jin, Y., Branke, J.: Evolutionary Optimization in Uncertain Environments – A Survey. IEEE Transactions on Evolutionary Computation 9(3), 303–317 (2005)
20. Jin, Y., Sendhoff, B.: A systems approach to evolutionary multiobjective structural optimization and beyond. IEEE Computational Intelligence Magazine 4(3), 62–76 (2009)
21. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (ICNN), pp. 1942–1948 (1995)
22. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publisher (2001)
23. Kulkarni, R.V., Venayagamoorthy, G.K.: Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 41(2), 262–267 (2011)
24. Liu, Y., Zhou, G., Zhao, J., Dai, G., Li, X.Y., Gu, M., Ma, H., Mo, L., He, Y., Wang, J., Li, M., Liu, K., Dong, W., Xi, W.: Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs. Frontiers of Computer Science in China 4(3), 334–338 (2010)
25. Lu, Y., Wang, S., Li, S., Zhou, C.: Particle swarm optimizer for variable weighting in clustering high-dimensional data. Machine Learning 82(1), 43–70 (2011)
26. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. Tech. rep., McKinsey Global Institute (May 2011)
27. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. Machine Learning 82(1), 1–42 (2011)
28. Pal, S.K., Talwar, V., Mitra, P.: Web mining in soft computing framework: Relevance, state of the art and future directions. IEEE Transactions on Neural Networks 13(5), 1163–1177 (2002)
29. Rajaraman, A., Leskovec, J., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press (July 2012)
30. Scott, D.W., Thompson, J.R.: Probability density estimation in higher dimensions. In: Gentle, J.E. (ed.) Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface, pp. 173–179 (1983)
31. Sheppard, J.W., Salzberg, S.L.: A Teaching Strategy for Memory-Based Control. Artificial Intelligence Review 11(1-5), 343–370 (1997)
32. Shi, Y.: An optimization algorithm based on brainstorming process. International Journal of Swarm Intelligence Research (IJSIR) 2(4), 35–62 (2011)
33. Slaney, M., Casey, M.: Locality-sensitive hashing for finding nearest neighbors. IEEE Signal Processing Magazine 25(2), 128–131 (2008)
34. Teodorović, D.: Transport modeling by multi-agent systems: A swarm intelligence approach. Transportation Planning and Technology 26(4), 289–312 (2003)

Multidimensional Dynamic Trust Measurement Model with Incentive Mechanism for Internetware

Guisheng Yin, Yingjie Wang, Jianguo Zhang, and Hongbin Dong

College of Computer Science and Technology,
Harbin Engineering University, Harbin, China
{yinguisheng,wangyingjie,zhangjianguo,donghongbin}@hrbeu.edu.cn

Abstract. The present trust measurement models of Internetware lack adaptability and time-sensitive, and cannot effectively deal with malicious behaviors. Therefore, based on characteristics of forgotten curve and information entropy theory, this paper proposes a multidimensional dynamic trust measurement model with incentive mechanism for Internetware. Experimental results show that this model can effectively restrain strategic dynamic changing behaviors of malicious nodes, and has good effectiveness and adaptability for different types of nodes.

Keywords: Internetware, trust measurement, time-sensitive, incentive mechanism.

1 Introduction

Internetware is an abstract of complex software in open networks. Internetware can perceive dynamic changes of external network environment. With the changes, entities adjust statically and evolve dynamically according to the network rules and indexes, so that make system have high trust degree [1]. As a new software type, Internetware and its technical system provide methods and technology foundation for researching trust and service quality of complex software [2].

In evolutionary process of Internetware system, people often expect system to converge to target state through effective methods and the selection of interactive strategies. Trust technology of network is a new security method based on network security technology. It strengthens dynamic process of network state and provides strategies for implementing adaptive network security and controlling service quality [3]. Trust is one of the most complex social relationships. It is an abstract psychological cognitive process. So it is hard to express and predict trust. In addition, because of the lack of corresponding incentive mechanism, 1/4 entities are free-rider, which leads to product free-riding phenomenon [4,5].The present research results about trust mechanism have effectively promoted the development of related work, and greatly enriched the understanding about trust relationship between entities. However, there are still some problems as follows.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 427–436, 2013.

(1) Most of the present models adopt expert or average weights method. Once weights are confirmed, it is hard to adjust them dynamically, so that makes trust measurement model lack adaptability.

(2) Most of the present models fail to consider time-sensitive of history behavior data, i.e., time decay problem about the history behavior data.

(3) Most of the present models cannot effectively deal with strategic dynamic changing behavior of malicious entities.

According to the above problems, based on forgetting curve of psychology and information entropy theory, this paper proposes a multidimensional dynamic trust measurement model with incentive mechanism for Internetware. Considering the time decay of history data, this paper introduces time decay factor to compute direct trust entities based on Ebbinghaus Forgetting Curve. In order to improve the accuracy of the recommendation results, this paper computes indirect trust of Internetware entities based on multi-path. In addition, an incentive mechanism is established to encourage entities to select high trust strategy. And a fluctuation-suppress parameter is introduced in order to overcome strategic dynamic changing behavior of malicious entities. In addition, this paper calculates the effectiveness of measurement results by computing evaluation similarity of entities.

2 Related Work

1994, Thomas Beth proposed trust measurement method under open network environment. He divided trust into direct trust and recommendation trust, and adopted probability method to signify trust. Wang Yuan [6] proposed a trust model based on Bayesian network. This model researches how to describe different aspects of trust, so that obtains various properties of entities according to different scenes. Wang Yao [7] solved the problem of recommendation based on Bayesian method. Lu Wen [8] proposed an evaluation method of software reliability. It is a bottom-up calculation process of trust level that can decompose parallel structure, so that calculate trust value of system accurately. However, it only adopts probability method to establish subjective trust model, so that cannot reflect real situations of trust relationship.

In software service synergic system based Agent, trust means a collaborative Agent that can predict subjective possibility of a collaborative activity. Prediction results are affected by evaluation of important degree [9,10]. Because trust between Agents associate with other entities' subjective understanding and fuzziness, it cannot be described and managed by conventional and accurate logic. Subjective trust as a cognitive phenomenon, its subjectivity and uncertainty present fuzziness, so scholars introduced Fuzzy Set into trust management. It not only reflects fuzziness of Agent trust, but also describes trust mechanism between Agents with intuitive and concise semantics. Tang Wen [11] proposed evaluative mechanism of trust based on fuzzy set theory. He gave formalization of trust, and deducted rule of trust to construct a complete subjective trust management model.

Dynamic trust model, one of the most complex social relationships in large-scale distributed systems, involves many factors. These factors are difficult to accurately express and predict. By combining rough-set theory with information entropy theory, Li Xiaoyong [12] put it in an open environment to dynamically build behavior-based data monitoring and analysis of trust measurement model. This method overcame the inadequate problem that traditional models deal with multi-dimensional data.

In addition, Xianfu Meng [13] proposed @Trust model. Puman Bedi [14] proposed trust based recommender system using ant colony for trust computation. And Zhang Shibin [15] proposed trust evaluation method based on cloud model. Above models has greatly promoted the research of trust measurement.

3 Dynamic Trust Measurement Model

By computing direct trust, indirect trust, incentive function and similarity, this paper measures trust degrees of Internetware entities comprehensively.

3.1 Multidimensional Direct Trust with Time Decay Factor (TDF-Trust)

Assume there are N entities in system: P_1, P_2, , P_N, $D_{i,j}^m$ is direct trust of the mth interaction between entity P_i and entity P_j.

$$D_{i,j}^m = \begin{cases} \frac{1}{m}\sum_{k=1}^{m} S_{i,j}^k & m \neq 0 \\ 0 & m = 0 \end{cases} \tag{1}$$

In Eq. (1), m is time-window. $S_{i,j}^{(k)}$ is satisfaction of interaction that entity P_i evaluates entity P_j in the kth interaction, and $S_{i,j}^{(k)} \in [0, 1]$.

In order to solve the problem of thicker dimension, and overcome the subjective judgment on weights, this paper divides satisfaction of interaction into four attributes: Reliability (Rt), Availability (At), Safety (St), and Maintainability (Mt). Thus, the calculation method of satisfaction of interaction that entity P_i evaluates entity P_j in the kth interaction is shown in Eq. (2).

$$S_{i,j}^k = \omega_1 \cdot Rt_{i,j}^{(k)} + \omega_2 \cdot At_{i,j}^{(k)} + \omega_3 \cdot St_{i,j}^{(k)} + \omega_4 \cdot Mt_{i,j}^{(k)} \tag{2}$$

In Eq. (2), ω is the corresponding weight of trust attribute, and $\omega_1 + \omega_2 + \omega_3 + \omega_4$. The corresponding weights represent the importance degrees of satisfaction of interaction. The influences that different trust attributes affect satisfaction of interaction are different. Information entropy theory is uncertainty of an event or measurement of information. It also can be understood as its own information included in the event. Based on information entropy theory, this paper evaluates corresponding weights of different trust attributes. Thus, the calculation method of information entropies for different trust attributes is shown in Eq. (3).

$$H(Xt_{i,j}^{(k)}) = -Xt_{i,j}^{(k)} \cdot log_2 Xt_{i,j}^{(k)} - (1 - Xt_{i,j}^{(k)}) log_2 (1 - Xt_{i,j}^{(k)}) \tag{3}$$

In Eq. (3), $Xt_{i,j}^{(k)}$ is the satisfaction degree of corresponding trust attribute. Let $C(Xt_{i,j}^{(k)})$ be the degree of differentiation that $Xt_{i,j}^{(k)}$ compared with other trust attributes. Its calculation method is shown in Eq. (4). And the calculation method of corresponding weight is shown in Eq. (5).

$$C(Xt_{i,j}^{(k)}) = \begin{cases} 1 - exp(-Sl) \cdot H(Xt_{i,j}^{k}) & Xt_{i,j}^{k} > 0.5 \\ 0 & Xt_{i,j}^{k} \leq 0.5 \end{cases} \tag{4}$$

$$\omega = \frac{C(Xt_{i,j}^{k})}{\sum_{Xt}^{Xt=Rt,At,St,Mt} C(Xt_{i,j}^{k})} \tag{5}$$

In Eq. (4), Sl is the service level that the entity has obtained in last interaction, and $Sl \in 1, 2, 3, 4, 5$. 1 represents the lowest service level, 5 represents the highest service level, and $0 \leq \omega \leq 1$. So we can compute satisfaction of interaction $S_{i,j}^{(k)}$ through Eq. (2) after confirming classified weights.

For improving accuracy and dynamic adaptability of trust measurement, and solving time-sensitive problem of history information, this paper introduces time decay factor $\lambda(k)$. It means the time decay factor in kth interaction, and $\lambda(k) \in (0, 1]$. In the latest interaction, time decay factor is equal to 1; conversely, in the farthest interaction, time decay factor closes to 0. According to characteristics of time decay factor, it can be seen that time decay factor follows the characteristics of Ebbinghaus Forgetting Curve. This curve shows a law of forgotten development. Forgotten process is not balanced. In initial stage, knowledge is soon forgotten, and then slowly. When it achieves some time, knowledge almost is not forgotten.

Internetware is the system with characteristic of ecological population. And memory of entities follows the forgotten development law of Ebbinghaus Forgetting Curve. Based on the forgotten development of Ebbinghaus Forgetting Curve and the law of forgotten development, this paper gives the calculation method of time decay factor, as shown in Eq. (6). The calculation method of direct trust with time decay factor is shown in Eq. (7).

$$\lambda(k) = \begin{cases} 1 & k = m \\ e^{-\frac{1}{k}} & 1 \leq k < m \end{cases} \tag{6}$$

$$D_{i,j} = \begin{cases} \frac{\sum_{k=1}^{m} D_{i,j}^{k} \cdot \lambda(k)}{\sum_{k=1}^{m} \lambda(k)} & m \neq 0 \\ 0 & m = 0 \end{cases} \tag{7}$$

3.2 Multi-Path Indirect Trust (MP-Trust)

In an Internetware system, interaction between entities not only has direct interaction behaviors, but also has indirect interaction behaviors. If there is an interaction path between two entities, there is trust relationship between them. Thus, this paper adopts the method of multi-path to compute indirect trust between entities.

Because the nearer between entities, more reliable the evaluation information, this paper computes indirect trust between entities through multi-path. This method is different from traditional calculation method of simple weighted average. It weights linked degree between entities to compute indirect trust of entities. Fig. 1 shows the multi-path diagram of trust interaction between entities. And $w_{i,j}$ represents trust weight between P_i and P_j.

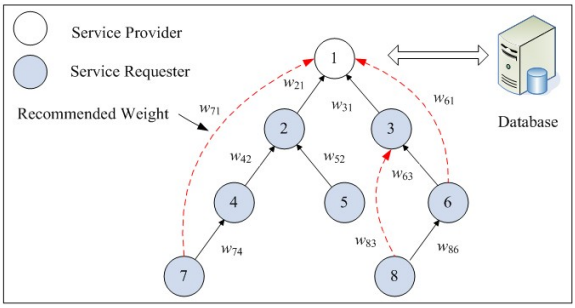


Fig. 1. Multi-path diagram of trust interaction between entities

Assume there is a path between P_i and P_j, $R_{i,j}$ represents indirect trust between P_i and P_j. $IL_{i,j}$ represents interaction path set between P_i and P_j. $IL^x_{i,j}$ represents the xth interaction path between P_i and P_j. $|IL_{i,j}|$ is the number of interaction paths between P_i and P_j. Thus, the calculation method of indirect trust between entities is shown in Eq. (8), and L_x is the nodes set on the xth interaction path between P_i and P_j. k and l are nodes on path L_x. $|L_x|$ is the number of nodes in the xth interaction path between P_i and P_j. Through successful interaction times, this paper confirms trust weight w between two entities. Assume the times of successful interaction between P_k and P_l are $I^{Success}_{k,l}$, and the total times of interaction times are $I^{Sum}_{k,l}$. So the calculation method of trust weight w is shown in Eq. (9).

$$R_{i,j} = \begin{cases} \dfrac{\sum_{IL^x_{i,j} \in IL_{i,j}}^{|IL_{i,j}|} \sum_{k,l \in L_x}^{|L_x|} w^x_{k,l} D^x_{k,l}}{|IL_{i,j}| \cdot \sum_{k,l \in L_x}^{|L_x|} w^x_{k,l}} & |L_x| \geq 2 \\ 0 & |L_x| = 0, 1 \end{cases} \tag{8}$$

$$w_{k,l} = \frac{I^{Success}_{k,l}}{I^{Sum}_{k,l}} \tag{9}$$

3.3 Global Trust

This paper gives the calculation method of global trust. Assume $M_{i,j}$ is the global trust, so the calculation method is shown in Eq. (10). ϖ_1 and ϖ_2 are respectively direct and indirect weights, and $\varpi_1 + \varpi_2 = 1$. In addition, $\varpi_1 = \frac{Q_{D_{i,j}}}{Q_{i,j}}$, $Q_{D_{i,j}}$

is the times of direct interaction between P_i and P_j, $Q_{i,j}$ is the total times of interaction between P_i and P_j, and $\varpi_2 = 1 - \varpi_1$.

$$M_{i,j} = \varpi_1 D_{i,j} + \varpi_2 R_{i,j} \tag{10}$$

3.4 Incentive Function with Fluctuation-Suppress Mechanism (FIM)

In Internetware system, the inherent ideal of entity is that it maximizes its profit, so as to induce distrust strategy to hold dominant position. Because total profit of distrust is greater than total profit of trust, selfish behavior dominates trust evolution direction of Internetware, which produces free-riding phenomenon. It makes distrust strategy be the first strategy for some entities.

In order to avoid free-riding phenomenon, let $I(j)$ be the incentive function, so the incentive function is shown in Eq. (11). $C(j)$ is the contribution of P_j for system, $E(j)$ is the payoff of P_j, α is the contribution factor, β is the profit factor, and $\alpha + \beta = 1$. $N_{i,j}$ is the fluctuation-suppress parameter. It overcomes strategic dynamic changing behavior of malicious nodes. Thus, the calculation method of fluctuation-suppress parameter is shown in Eq. (12).

$$I(j) = \alpha C(j) + \beta E(j) - N_{i,j} \tag{11}$$

$$N_{i,j} = \left| M_{i,j}^{(n)} - M_{i,j}^{(n-1)} \right| \tag{12}$$

3.5 Similarity

In order to determine effectiveness that P_i measures other entities, this paper depicts effectiveness of trust measurement through similarity. Let $Similar_{i,r}$ be the similarity of trust measurement between P_i and P_r. The bigger similarity between them, the more trust evaluation of P_i. The similarity is calculated by Eq. (13). And $set(i, r)$ is the entity set that have interacted with P_i and P_j. Thus, for computing trust of P_j, it must been determined evaluation effectiveness of P_i. The comprehensive trust of entities is shown in Eq. (14).

In order to normalize the comprehensive trust of Internetware entities, this paper defines a dimensionless $t_{i,j}$, and $t_{i,j} = \frac{T_{i,j}}{T_0}$. T_0 is a system constant, and $0 \leq T_{i,j} \leq T_0$.

$$Similar_{i,r} = \frac{\sum_{k \in set(i,r))} M_{i,r} M_{r,k}}{\sqrt{\sum_{k \in set(i,r))} M_{i,k}^2} \times \sqrt{\sum_{k \in set(i,r))} M_{r,k}^2}} \tag{13}$$

$$T_{i,j} = \frac{\sum_{q=1}^{n} Similar_{i,q}}{n} \times M_{i,j} + I(j) \tag{14}$$

4 Experiments and Result Analysis

According to the above analysis, this paper gives the experiments and result analysis for this model, and verifies its effectiveness and adaptability.

The hardware experimental environment is Intel Core(TM) Duo 2.66GHz CPU, 2GB memory, Windows XP operating system and Matlab 7.0 experimental platform.

4.1 Effectiveness of Resist Malicious Fluctuation

The ability that system resists malicious fluctuation of entities is verified in this section. This paper respectively simulates dynamic trust changing of malicious entities, which fluctuating periods are 30 and 20 generations, as shown in Fig. 2. Through comparing with PeerTrust model, this paper verifies the effectiveness that model resists malicious fluctuation behavior of entities. In figures, red line shows the fluctuation behavior of malicious entities. TDF is the trust measurement process with time decay factor. FIM is the trust measurement process with incentive function. TDF+FIM is the trust measurement process with time decay factor and incentive function.

From Fig. 2, it can be seen that PeerTrust model and our model both can sense the changes of entities in time. However, PeerTrust cannot reflect time-sensitive of history trust data and resist malicious fluctuation behavior of entities. And the trust model without any trust mechanism cannot resist malicious fluctuation behavior of entities. TDF has certain inhibiting effect. And it reflects time-sensitive of history trust data. FIM reduces trust degree after the malicious fluctuation of entities. But its effect is not obvious, and the fluctuation is big still. TDF+FIM has a good inhibitory effect for malicious fluctuation behavior. It reflects time-sensitive characteristic of history trust data. And trust degree of entity is at a low level after fluctuating, so it verifies effectiveness of this model that resists malicious fluctuation behavior of entities.

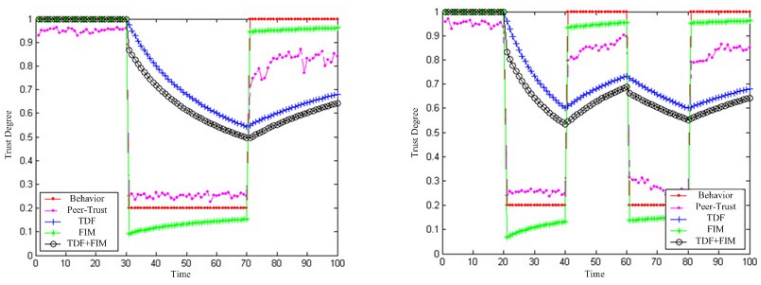


Fig. 2. Dynamic trust changing process of malicious entities

4.2 Effectiveness and Adaptability of Model

For verifying effectiveness and adaptability of this model, this paper simulates trust changing process of normal entities. Fig. 3 shows the experimental result of trust changing process of normal entities. In Fig. 3, the trust changing ranges are respectively 0.6-0.8 and 0.2-1.0. From Fig. 3, it can be seen that when trust fluctuates in certain range, FIM has certain punishment effect for trust fluctuation behavior, but cannot reflect time-sensitive characteristic of history trust information. TDF reflects time-sensitive characteristic of history trust information, and has a good stability. But the punishment force for fluctuation behavior is not very obvious. TDF+FIM not only has better stability, but also has a certain inhibitory effect for fluctuation behavior of malicious entities.

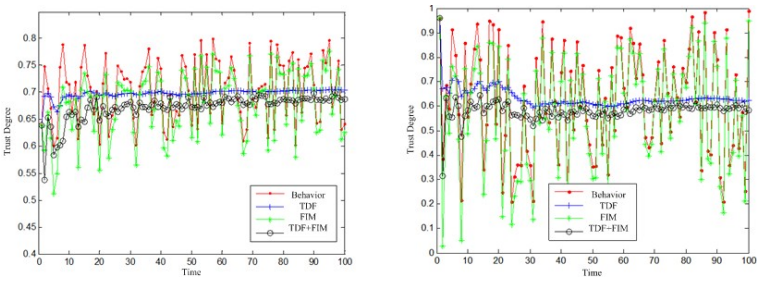


Fig. 3. Dynamic Trust Changing Process of Normal Entities

This paper also measures the trust degree for lazy and learning entities. So-called lazy entities are the nodes with normal learning ability. However, after some interactions, the trust degree of entities always at a low level. And learning entities are the nodes with good learning ability. Learning entities can improve their own trust continually, until reach the stable state. The left figure of Fig. 4 shows the dynamic trust changing process of lazy entities. It can be seen that TDF+FIM has a certain inhibitory effect for fluctuation of entities under situation that initial trust of entities grow fast. And in lazy stage, TDF+FIM does not decrease its trust degree fast. The right figure of Fig. 4 shows the dynamic trust changing process of learning entities. TDF+FIM has a certain inhibitory effect for initial trust fluctuation. But when trust behavior tends to the stable state, TDF+FIM gradually improves the trust degree of entities and tends to the stable state. And compared with TDF, when the trust degree of entities tends to a stable state, trust behaviors of entities have little fluctuation. TDF+FIM can improve the trust degree of entities fast, so it has a good adaptability. Thus, this dynamic trust measurement model has good effectiveness and adaptability for different types of entities.

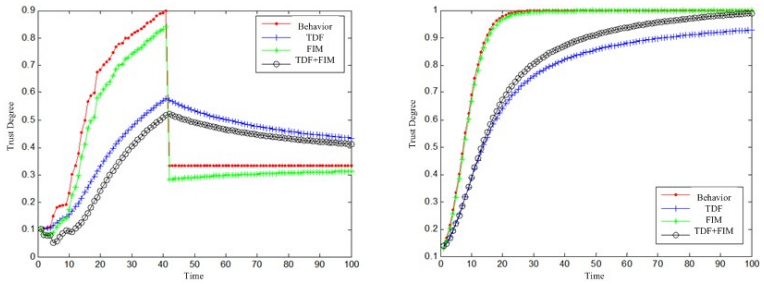


Fig. 4. Dynamic trust changing process

5 Conclusions

According to the shortages of present trust measurement models, most of them are single dimension and fail to consider time-sensitive of history trust data. In addition, they cannot effectively deal with strategic dynamic changing behavior of malicious entities. A multidimensional dynamic trust measurement model with incentive mechanism for Internetware is proposed based on the characteristic of forgetting curve and information entropy theory. This paper proposes multidimensional direct trust with time decay factor, indirect trust based on multi-path indirect trust, incentive function with fluctuation-suppress mechanism and similarity. Through experiments, we can see that this model can effectively inhibit strategic dynamic changing behavior of malicious entities. And this paper verifies that this model has good effectiveness and adaptability for different types of entities.

However, this paper does not consider other trust behaviors and network environment factors. Therefore, the further research on complex trust relationships between Internetware entities is the focus of the future research.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No.61272186 and No.61100007, the Natural Science Foundation of Heilongjiang Province under Grant No.F201110, the Fundamental Research Funds for the Central Universities under Grant No.HEUCF100608, and the Foundation of Heilongjiang Postdoctoral under Grant No.LBH-Z12068.

References

1. Lv, J., Ma, X., Tao, X., Xu, F., Hu, H.: Research and Development of Internetware. Science in China 36, 1037–1080 (2006)
2. Yang, F.Q.: Thinking on the Development of Software Engineer. Journal of Software 16, 1–7 (2005)

3. Wang, P., Sun, C., Li, L.: Primary Research on Internetware Reliability Technology. In: First International Multi- Symposiums on Computer and Computational Sciences, IMSCCS 2006, pp. 424–428. Institute of Electrical and Electronics Engineers Computer Society, Hangzhou (2006)
4. Yue, G., Li, R., Chen, Z., Zhou, X.: Analysis of Free-riding Behaviors and Modeling Restrain Mechanisms for Peer-to-Peer Networks. Journal of Computer Research and Development 48, 382–397 (2011)
5. Jun, S., Ahamad, M.: Incentives in BitTorrent Induce Free Riding. In: Proceedings of the ACM SIGCOMM. Workshop on Economics of Peer-to-Peer Systems, P2P3CON, pp. 116–121. Association for Computing Machinery, Philadelphia (2005)
6. Wang, Y., Lv, J., Xu, F., Zhang, L.: A Trust Measurement and Evolution Model for Internetware. Journal of Software 17, 1–2 (2006)
7. Wang, Y., Vassileva, J.: Bayesian network-based trust model. In: Proc. of the IEEE Computer Society WIC International Conference on Web Intelligence, pp. 372–378. IEEE Computer Society, Washington DC (2003)
8. Lu, W., Xu, F., Lv, J.: An Approach of Software Reliability Evaluation in the Open Environment. Chinese Journal of Computers 33, 452–462 (2010)
9. Zhu, M., Jin, Z.: Approach for Evaluating the Trustworthiness of Service Agent. Journal of Software 22, 2593–2609 (2011)
10. Chang, Z., Mao, X., Qi, Z.: Component Model and Its Implementation of Internetware Based on Agent. Journal of Software 19, 1113–1124 (2008)
11. Wen, T., Zhong, C.: Research of Subjective Trust Management Model Based on the Fuzzy Set Theory. Journal of Software 14, 1401–1408 (2003)
12. Li, X., Gui, X.: Trust Quantitative Model with Multiple Decision Factors in Trusted Netware. Chinese Journal of Computers 32, 405–416 (2009)
13. Xianfu, M., Yalin, D., Yue, G.: @Trust: A trust model based on feedback-arbitration in structured P2P network. Computer Communications 35, 2044–2053 (2012)
14. Bedi, P., Sharma, R.: Trust based recommender system using ant colony for trust computation. Expert Systems with Applications 39, 1183–1190 (2012)
15. Zhang, S., Xu, C.: Study on the Trust Evaluation Approach Based on Cloud Model. Chinese Journal of Computers 36, 422–431 (2013)

Global Path Planning of Wheeled Robots Using a Multi-Objective Memetic Algorithm

Fangxiao Wang and Zexuan Zhu*

Shenzhen City Key Laboratory of Embedded System Design,
College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen 518060, China
zhuzx@szu.edu.cn

Abstract. This paper proposes a multi-objective memetic algorithm (MOMA) for global path planning of wheeled robots. Particularly, MOMA is designed to simultaneously optimize the path length and smoothness. MOMA is featured with novel path encoding scheme, path rectification, and specific evolutionary operators. The experimental results on simulated maps show that MOMA is efficient in planning a set of valid trade-off paths in complex environments.

Keywords: Multi-objective optimization, Global path planning, Memetic algorithm.

1 Introduction

Mobile robots have been widely used in various areas such as planet exploration, surveillance, and landmine detection. As one key issue of robotics, path planning aims to determine a collision-free path from a start location to a target location while optimizing some performance criteria like distance, time consumption, and energy cost. Path planning has attracted remarkable attention and many encouraging results have been achieved.

Traditional path planning methods including roadmap methods, cell decomposition methods, and potential field methods have been successfully applied to many path planning problems, whereas they are likely time consuming and/or tend to get trapped in local optimal in complex problems [1]. Evolutionary algorithms (EAs) such as genetic algorithms [2], ant colony optimization [3], and particle swarm optimization [4] server as another type of solution for path planning. EAs have achieved innovative achievements in path planning, making up for the drawback of traditional methods.

Most path planning techniques mentioned above are used to solve single objective optimization problem. However, in real-world applications, there usually have more than one optimization objectives, e.g., in the path planning of wheeled robots, the length, smoothness, and validity of a path are the most commonly considered objectives. Multi-objective optimization based path planning methods are demanded to find the trade-off solutions. Many researchers have proposed to use multi-objective EAs for solving path planning. For example, Hu *et al*. proposed a multi-objective mobile robot

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 437–444, 2013.

path planning based on an improved genetic algorithm [5]. The algorithm optimizes three objectives, i.e., the length, smoothness, and security of a path. Gong *et al.* [6] presented a multi-objective particle swarm optimization for robot path planning in environment of danger sources. Two objectives namely the length and the danger extent of the path were considered. Mo *et al.* [7] introduced a constrained multi-objective biogeography optimization algorithm for optimizing the path length and smoothness simultaneously, and the degree of a path blocked by obstacles was imposed as a constraint. Hung *et al.* introduced the second objective, penetration-depth of obstacles, besides the path length, for path planning in [8].

In this study, a multi-objective memetic algorithm (MOMA) [9,10] is proposed for the global path planning of wheeled or car-like robots. MOMA is designed to optimize the path length and smoothness, simultaneously. In many real-world problems, the two objectives are conflicting with each other and no preference should be imposed on either one of them. The path validity is handled by a novel path rectification operation in MOMA. Specific path encoding scheme and evolutionary operators are also introduced to solve the global path planning problem. The performance of MOMA is evaluated using three simulated maps. The experimental results show that MOMA is able to obtain a set of Pareto optimal paths efficiently, which provides great conveniences for choosing appropriate path according to different preferences in real-world practices.

The remainder of this paper is organized as follows. Section 2 describes the formulation of path planning problem and the path representation. The proposed MOMA is presented is Section 3. Comparison and stimulation studies are provided in Section 4. Finally, conclusions are given in Section 5.

2 Problem Formulation and Path Representation

Depending on whether full knowledge of the environment is available, path planning can be categorized into global path planning and local path planning. In this study, we focus on global path planning in which the complete information of the environment is known in advance. Given an environment $E(S, T, \mathbf{O})$ where S is the starting point, T is the target point, and $\mathbf{O}$ denotes the set of obstacles, global robot path planning aims to find a collision-free path from S to T while optimizing some criteria. In this study, the map of the environment is constructed based on C-space [11], where the mobile robot is presented as a single point and the obstacles are represented as polygons. Since the robot is shrunk to a point, the obstacles are enlarged by the size of the robot for compensation (as shown in Fig. 1). The obstacle polygons and their vertices are all indexed with numbers.

In MOMA, a candidate path is represented as a sequence of obstacles gone around by the path. As such, the search of candidate paths is constrained in the space of obstacles. Particularly, a variable length integer chromosome encoding scheme shown in Fig. 2 is used. The chromosome consists of n 4-tuple genes with each gene $\{O^i, V^i, N^i, D^i\}$ indicating a bypassed obstacle, where O^i denotes the index of the obstacle, V^i indicates the index of the entry vertex from which the path circuits the obstacle, N^i represents the number of vertices the path goes around the obstacle, and D^i =0(1) denotes that the path bypasses the obstacle in clockwise (counter-clockwise) direction.

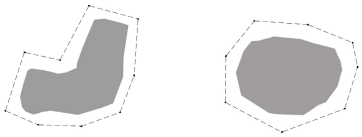


Fig. 1. Obstacles enlarged by the size of the robot

Fig. 2. Chromosome encoding scheme

To construct a path from a chromosome, the starting point S, the corresponding vertices of the encoded obstacles, and the target point T are sequentially connected with straight-line segments. As an example shown in Fig. 3, a chromosome [1,1,2,1;3,1,2,0] of two genes indicates that the encoded path starts from S and circuits obstacle 1 from the first vertex in counter-clockwise direction for two vertices, then goes around obstacle 3 from the first vertex in clockwise direction for two vertices, i.e., from the first vertex to the fourth vertex, and finally reaches T.

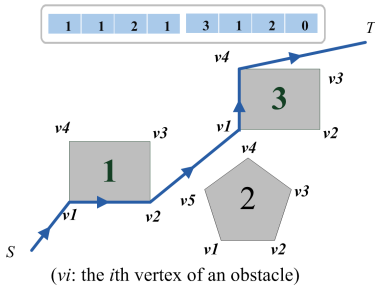


Fig. 3. Path construction based on a chromosome

3 Multi-Objective Memetic Algorithm for Global Path Planning

In recent years, the idea of Pareto-optimality is introduced to solve multi-objective optimization problem with the advantage that multiple trade-off solutions can be obtained in a single run. Without loss of generality, when minimizing objectives $f_1, f_2, \ldots, f_K$, a feasible solution P_i is said to dominate another feasible solution P_j, if and only if $f_k(P_i) \leq f_k(P_j)$ for $k = 1, \ldots, K$. A Pareto optimal solution is a solution not dominated by any other solution. The set of all Pareto optimal solutions in the solution space is referred to as the Pareto optimal set. The corresponding objective function values of all Pareto optimal solutions in the objective space form the Pareto front.

Algorithm 1. Procedure of MOMA

1. Randomly initialize the population $\mathbf{P}_0$ and set $g = 0$;
2. Apply path rectification on all individuals;
3. **while** stopping criteria are not satisfied **do**
4. Evaluate the fitness of each individual based on Equations (1)-(3);
5. Generate an offspring population $\mathbf{Q}_g$ with selection, crossover, and mutation;
6. Apply path rectification on the offspring individuals;
7. Combine $\mathbf{P}_g$ and $\mathbf{Q}_g$
8. Generate next population $\mathbf{P}_{g+1}$ based on nondomination rank and crowding distance [12];
9. $g = g + 1$;
10. **end while**

Based on the problem formulation, path representation, and the definition of Pareto-optimality, we propose the multi-objective memetic algorithm (MOMA) for global path planning. The outline of MOMA is show in Algorithm 1 and the details are provided in the following subsections.

3.1 Path Rectification

In each evolution generation of MOMA, the individuals undergo path rectification to ensure the validity of the encoded paths. The path rectification simply checks the path segment by segment from S to T. If a straight-line segment is detected to collide with an obstacle which is not involved in the path, the collided obstacle is inserted to the path. If the collided obstacle has already been encoded, the rectification operation changes the values of V^i, N^i and/or D^i of the corresponding gene to avoid the collision.

To illustrate the procedure, an example is given in Fig. 4. In Fig. 4(a), the candidate path collides with obstacles 1 and 4. In Fig. 4(b), the path rectification firstly detects the collision in the first straight-line segment and inserts obstacle 1 to the path by detouring it in clockwise direction. Afterward, the second collision is detected on obstacle 4 which has been included in the path, so the rectification just modifies the entry vertex V^i to obtain a valid solution as shown in Fig. 4(c).

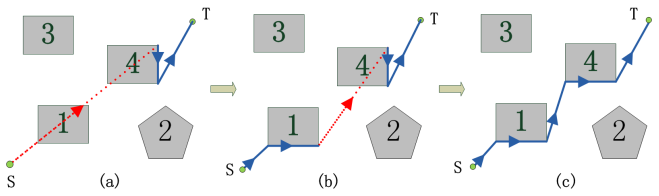


Fig. 4. An example of path rectification

3.2 Fitness Evaluation

In the path planning of wheeled robots, the cost of robots' traveling is mainly evaluated in terms of path length, path smoothness, and the number of collisions. In MOMA,

the collision issue is taken care in the path rectification, leaving the path length and smoothness as the two main objectives to optimize.

Given a candidate path $P = \{p_1, p_2, \ldots, p_n\}$, where $p_1 = s$ is the start point, $p_n = t$ is the target point, and $p_i (i = 2, \ldots, n-1)$ denotes the i-th intermediate node of the planned path P. The multi-objective optimization in MOMA is simplified by finding an optimal path P^* such that:

$$f(P^*) = \min_{P^* \in P} \{f_1(P), f_2(P)\} \tag{1}$$

where $f_1(P)$ and $f_2(P)$ are the objective functions of path length and smoothness, respectively. $f_1(P)$ is defined as follows:

$$f_1(P) = \sum_{i=1}^{n-1} |p_i p_{i+1}| \tag{2}$$

where $|p_i p_{i+1}|$ denotes the Euclidean distance between p_i and p_{i+1}. The length of the path is the summary of the distance of every two sequential nodes.

The evaluation of path smoothness is estimated by the accumulated turning angle of the path as follows:

$$f_2(P) = \sum_{i=1}^{n-2} \theta(\overrightarrow{p_i p_{i+1}}, \overrightarrow{p_{i+1} p_{i+2}}) \tag{3}$$

where $\theta(\overrightarrow{p_i p_{i+1}}, \overrightarrow{p_{i+1} p_{i+2}}) (0 \leq \theta \leq \pi)$ is the vectorial angle between straight-line segments $\overrightarrow{p_i p_{i+1}}$ and $\overrightarrow{p_{i+1} p_{i+2}}$.

3.3 Evolutionary Operators

In each generation, MOMA evolves the individual population $\mathbf{P}_g$ using the evolutionary operators, i.e., selection, crossover, and mutation.

- Selection: A mating pool of size $|\mathbf{P}_g|$ is formed by random sampling of $\mathbf{P}_g$ with replacement. The min-max method [8] is used to guarantee the survival of extreme solutions on the Pareto front.
- Crossover: Every two parent chromosomes in the mating pool undergo one-point crossover in a predefined probability to generate two offspring chromosomes. Note that the crossover point should be in between the genes.
- Mutation: after crossover, the offspring chromosomes undergo mutation based on a dynamic mutation rate $m_g = (1 - 0.75g^2/G^2)m_0$, where g indicates the generation number, G is the total number of generations, and m_0 denotes the initial mutation rate. The components of a gene V^i, N^i and D^i are randomly changed in the corresponding ranges based on m_g. We also enable the mutation operator to randomly add/delete obstacles to the encoded path under the same probability. After crossover and mutation, path rectification is applied on the two offspring chromosomes to ensure their validity.

MOMA applies the evolutionary operators to generate an offspring population $\mathbf{Q}_g$ whose size is the same as $\mathbf{P}_g$. The parent population $\mathbf{P}_g$ and offspring population $\mathbf{Q}_g$ are combined together. Like NSGA-II [12], all the individuals are sorted by their nondomination rank and crowding distance. The half of the top ranked individuals are retained to form a new population $\mathbf{P}_{g+1}$ and a new generation begins. The above evolution procedure is repeated until some stopping criteria are satisfied.

4 Experimental Results

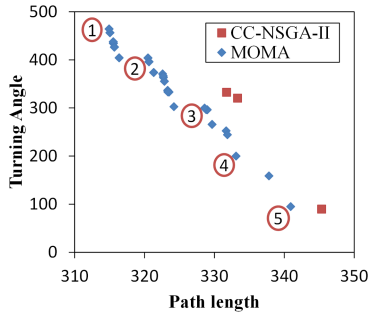
The performance of MOMA is tested using three simulated maps in this section. For comparison study, a colliding constrained NSGA-II (CC-NSGA-II) is considered. The original NSGA-II has no mechanism to handle invalid paths. Therefore, in addition to the path length and smoothness, a third objective, i.e., the number of collisions occurring in the segments of a path, is introduced to NSGA-II and form the CC-NSGA-II. CC-NSGA-II uses the same evolutionary operators as NSGA-II and its chromosome encoding scheme is the same as MOMA. The parameters of the two algorithms are empirically set to: generation number=50, population size=100, and crossover probability=0.65. The mutation rates of the two algorithms are set differently. Because of the use of dynamic mutation rate, MOMA applies a higher initial mutation rate of 0.3, while the mutation rate of CC-NSGA-II is set to 0.15. The results of the two algorithms are reported over 10 independent runs.

The best Pareto front obtained by MOMA and CC-NSGA-II in the 10 runs are plotted in Fig. 5. It can be seen that the Pareto optimal solutions of MOMA dominate the counterpart solutions of CC-NSGA-II in Maps 1 and 2. In Map 3, CC-NSGA-II fails to obtain valid solutions due to the complexity of the environment.

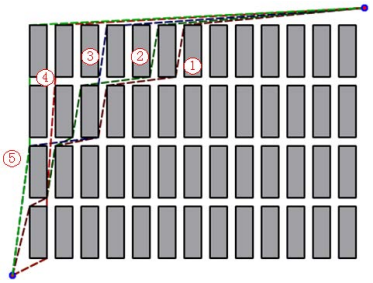
We also summarize the average size of Pareto front, the average number of feasible solutions in Pareto front, the average minimum length of the paths, and the average minimum turning angle of the paths obtained by the two algorithms in Table 1. It is shown that MOMA outperforms CC-NSGA-II in the three maps in terms of Pareto front size, percentage of obtaining valid Pareto optimal solutions, and the average minimum length of the path. With respect to the average minimum turning angle, CC-NSGA-II wins in Map 1 and is competitive to MOMA in Map 2, but fails to obtain valid solution in Map 3.

Table 1. Statistic of the Pareto fronts obtained by MOMA and CC-NSGA-II

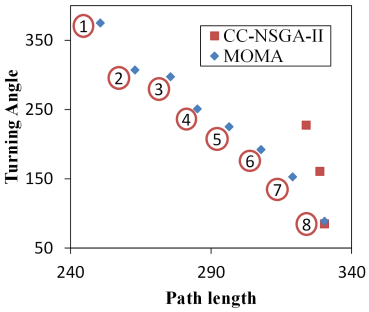
	Algorithm	Pareto front size (valid)	Min length	Min turning angle
Map1	MOMA	**52.9 (52.9)**	**317.627**	205.075
	CC-NSGA-II	4 (1.78)	340.428	**138.308**
Map2	MOMA	**23 (23)**	**267.087**	**118.503**
	CC-NSGA-II	5 (1.625)	327.973	124.745
Map3	MOMA	**12.1 (12.1)**	**409.873**	**198.706**
	CC-NSGA-II	2.2 (0)	–	–



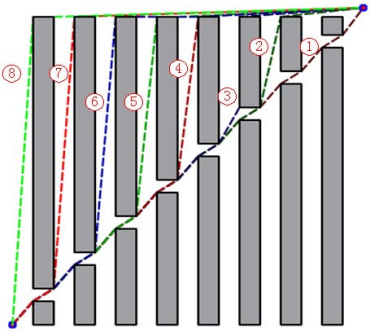
(a) Parato optimal solutions on Map 1



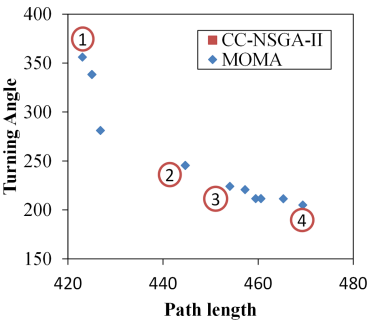
(b) Representative paths on Map 1



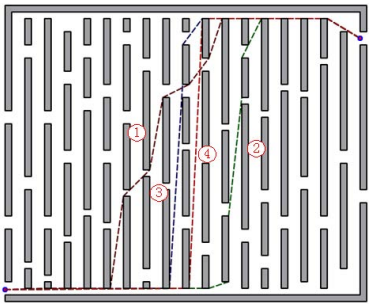
(c) Parato optimal solutions on Map 2



(d) Representative paths on Map 2



(e) Parato optimal solutions on Map 3



(f) Representative paths on Map 3

Fig. 5. Results on the simulated maps

5 Conclusion

In this paper, a multi-objective memetic algorithm (MOMA) is proposed to optimize the path length and smoothness of global path planning for wheeled robots. In MOMA, a variable length integer chromosome encoding scheme is proposed to represent the candidate paths and specific evolutionary operators are accordingly designed to fulfil the evolution. A novel path rectification mechanism is also introduced to guarantee the

validity of the paths obtained by MOMA. The experimental results on simulated maps demonstrate the efficiency of MOMA in complex environments.

Acknowledgements. This work was supported in part by the NSFC funding under grants 61001185 and 61205092, in part by the NSFC-RS joint project under grants IE111069 and 61211130120, in part by the Guangdong Natural Science Foundation, under grants S2012010009545 and S2012040007668, in part by Foundation for Distinguished Young Talents in Higher Education of Guangdong, China, under grants 2012LYM0116, in part by Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China, under grand 20111568, and in part by Science Foundation of Shenzhen City under grants JC201105170650A, KQC201108300045A, and JCYJ20130329115450637.

References

1. Raja, P., Pugazhenthi, S.: Optimal path planning of mobile robots: a review. International Journal of Physical Sciences 7(9), 1314–1320 (2012)
2. Holland, J.: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge (1992)
3. Dorigo, M.: Optimization, learning and natural algorithms. Ph.d thesis, Politecnico di Milano, Italy (1992)
4. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE (1995)
5. Jun, H., Qingbao, Z.: Multi-objective mobile robot path planning based on improved genetic algorithm. In: 2010 International Conference on Intelligent Computation Technology and Automation, vol. 2, pp. 752–756. IEEE (2010)
6. Gong, D.W., Zhang, J.H., Zhang, Y.: Multi-objective particle swarm optimization for robot path planning in environment with danger sources. Journal of Computers 6(8), 1554–1561 (2011)
7. Mo, H., Xu, Z., Tang, Q.: Constrained multi-objective biogeography optimization algorithm for robot path planning. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part I. LNCS, vol. 7928, pp. 323–329. Springer, Heidelberg (2013)
8. Hung, K.T., Liu, J.S., Chang, Y.Z.: A comparative study of smooth path planning for a mobile robot by evolutionary multi-objective optimization. In: 2007 International Symposium on Computational Intelligence in Robotics and Automation, pp. 254–259. IEEE (2007)
9. Moscato, P.: Memetic algorithm: A short introduction. McGraw-Hill, London (1999)
10. Zhu, Z., Ong, Y.-S.: Memetic algorithms for feature selection on microarray data. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007, Part I. LNCS, vol. 4491, pp. 1327–1335. Springer, Heidelberg (2007)
11. Lozano-Perez, T.: Spatial planning: a configuration space approach. IEEE Transactions on Computers 100(2), 108–120 (1983)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)

Quantifying Flow Field Distances Based on a Compact Streamline Representation

Lars Graening[1] and Thomas Ramsay[2]

[1] Honda Research Institute Europe,
Carl-Legien-Straße 30, 63073 OffenbachMain, Germany
lars.graening@honda-ri.de
[2] Honda R&D Americas, Inc, Raymond, OH 43067
tramsay@oh.hra.com

Abstract. In many engineering domains like aerospace, vehicle or engine design the analysis of flow fields, acquired from computational fluid dynamics (CFD) simulations, can reveal important insights on the behavior of the simulated objects. However, the huge amount of flow data produced by each simulation complicates the data processing and limits the application of computational tools for flow analysis. Thus, an a priori transformation of the flow data into a compact low dimensional representation is desired. This paper introduces a new procedure for transforming flow field data into a compact streamline based representation. Wherein, streamlines with negligible information contribution are removed from the representation. The reduced set of streamlines defines the basis for a subsequent quantification of flow field distances. Experimental studies show that the distances calculated based on the compact representation well approximate the distances of the uncompressed flow field with a significant drop in memory consumption.

Keywords: Flow field distance, Flow features, Compact flow field representation, CFD.

1 Introduction

The computational simulation and modeling of fluid dynamics has become a corner stone for a huge variety of applications, e.g. related to aerospace, climate modeling, vehicle or engine design. The analysis and comparison of the acquired flow field data can provide relevant domain information related to the behavior of the technical system under consideration, e.g. revealing important interactions between design and flow features [6]. The comparison of two flow fields requires a proper similarity measure being defined, which can e.g. be adopted for discovering flow field categories [1] or to compare simulated against ideal flow, as for the design of combustion engines [3]. However, means for the computational evaluation of flow field similarities are rarely being used in practice, rather modern visualization techniques [5] are applied to support the qualitative evaluation of flow field similarities through visual inspection. Quantitative methods require

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 445–454, 2013.

that explicit considerations about the flow field representation and the similarity measure are being made. Xu et al. [9] have studied probabilistic similarity measures based on featured flow fields. The measures take feature data from the entire simulation domain into account. For practical applications, this requires a big data storage and is linked to high computational costs for the similarity evaluation. In this paper a new compact streamline vise flow field representation is introduced which is derived from an automatic streamline reduction process. The reduced flow field representation is the basis for the definition of a probabilistic similarity measure.

2 Flow Field Representation

Flow fields, as a result of computational fluid dynamics simulations (CFD), define properties of the fluid or air flow in interaction with solid objects. Beside pressure, density and temperature, the velocity vector field defines the speed of the air or fluid flow at fixed positions in the Euclidean space. Considering steady state flow, the vector field is formally defined as a function that maps each point in the Euclidean space to a velocity vector, which remains constant over time,

$$v(p) : (x, y, z)^T \mapsto (v_x, v_y, v_z)^T.$$

Tools for computational flow simulation model the flow physics with the Navier Stokes equations. In order to numerically solve these partial differential equations the simulation area is subdivided into discrete cells. Especially for high fidelity simulations the cell sizes need to be kept small, resulting in a huge velocity vector field representing the detailed flow of the entire simulation domain. The storage of all the flow data is often impractical.

Local Flow Features. The orientation and magnitude of the velocity vector does not always provide all relevant information about the flow at a fixed point p. Therefore, additional features need to be derived. Wherein, the calculation of the features are always problem and application dependent. Following Xu et al. [9], features representing higher order properties derived from the gradient of the velocity vector field have been considered in this paper. The local gradient in a 3D velocity vector field is defined by an asymmetric 3D tensor:

$$T(p, v) = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix} = S + A, \tag{1}$$

which can be decomposed into its symmetric and asymmetric components, S and A:

$$S = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{1}{2}\left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right) & \frac{1}{2}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right) & \frac{\partial v_y}{\partial y} & \frac{1}{2}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) \\ \frac{1}{2}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) & \frac{\partial v_z}{\partial z} \end{pmatrix} = \begin{pmatrix} \varepsilon_1 & \frac{1}{2}\theta_3 & \frac{1}{2}\theta_2 \\ \frac{1}{2}\theta_3 & \varepsilon_2 & \frac{1}{2}\theta_1 \\ \frac{1}{2}\theta_2 & \frac{1}{2}\theta_1 & \varepsilon_3 \end{pmatrix}, \tag{2}$$

$$A = \begin{pmatrix} 0 & -\frac{1}{2}\left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right) & \frac{1}{2}\left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right) & 0 & -\frac{1}{2}\left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right) \\ -\frac{1}{2}\left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right) & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \tag{3}$$

where S comprises isotropic scaling and rotation information, and A anisotropic information about the stretching of the velocity vectors in the field, based on which the following scalar features can be calculated:

1. *Vector magnitude*: $M_v = |(v_x, v_y, v_z)^T|$,
2. *Tensor magnitude*: $M_t = \sqrt{\frac{1}{2}\sum_i^3 \sum_j^3 T_{i,j}^2}$,
3. *Dilatation magnitude*: $M_d = \sqrt{\sum_i^3 \varepsilon_i^2}$,
4. *Magnitude of shear strain rate*: $M_s = \sqrt{\sum_i^3 \theta_i^2}$,
5. *Magnitude of vorticity*: $M_\omega = \sqrt{\sum_i^3 \omega_i^2}$.

Given those five features we define the feature space as:

$$f(p, v) : p, v \mapsto (M_v, M_t, M_d, M_s, M_\omega)^T.$$

The defined feature vector is considered to comprise all problem relevant information within the simulated flow domain. If this is not the case any subsequent analysis and modeling step might fail to reveal desired information.

Featured Streamlines. Streamlines are one of the most important flow field visualization techniques see e.g. [10]. Streamlines describe the trajectories (tangential to the velocity vector) that particles would travel through the flow starting from an initial seeding position. Streamlines are represented by discrete sample points along the trajectory, where the distance between adjacent sample points is defined by the density of the CFD mesh. Assigning the feature vector to each sample point along the streamline, a *featured streamline* is defined as:

$$l(u) : u \mapsto p, v, (M_v, M_t, M_d, M_s, M_\omega)^T,$$

where u defines the position of the sampling point along the streamline l. A set of featured streamlines is denoted as $L = \{l_i\}, i \in [1, N_l]$, with N_l stream lines.

Beside the purpose of visualization, the transformation of the flow field into a set of streamlines can be applied for data reduction, where the entire flow and feature data is condensed into information along individual trajectories. Without a priori knowledge about the flow field, streamlines are seeded uniformly at the inlet. In order to capture all relevant flow phenomena, often a huge number of streamlines is required, what complicates the subsequent visual investigation of the flow and goes along with an increase in memory consumption.

3 Automatic Streamline Reduction

A program for automatic streamline reduction has been developed to reduce the amount of flow data to be processed by humans or computer programs. Given a large set of streamlines, the algorithm targets to remove individual streamlines from the set which contribute least to the overall information content. Thus, targeting the creation of an efficient streamline based flow field representation with a minimum number of streamlines. Therefore, streamlines are ranked according to their information contribution, which is quantified based on the formulation of the Shanon entropy [8]. Given discrete feature data for each streamline, the expected entropy accumulated over all features defines the considered measure of relevance, formally defined as:

$$\overline{H}_i = E(H(l_i)) = -\frac{1}{N_f} \sum_{i}^{N_f} \sum_{j}^{N_b^i} p(\hat{f}_i^j) \log p(\hat{f}_i^j),$$

with N_f defining the number of features, N_b^i defining the number of discrete states and $\hat{f}_i^j$ defining the discrete feature value of feature i, e.g. the velocity magnitude M_v. The probability $p(\hat{f}_i^j)$ denotes the likelihood of a discrete feature value to be measured along the considered streamline. On the one hand, if all discrete feature values are equally probable, the feature values are uniformly distributed and the entropy gets maximal. On the other hand, if $p(\hat{f}_i^j) = 1.0$, the feature value along the streamline does not change within a certain boundary. In the latter case the value of the Shannon entropy vanishes. Under the assumption that streamlines with none or small feature value variations are of less relevance and capture least information about the flow field, a heuristic has been defined that removes streamlines with smallest $\overline{H}_i$. Ranking all streamlines according to their expected entropy, a subset of streamlines L' can be derived which captures the majority of the overall flow feature information. The information content of the reduced subset is defined by the relative expected entropy:

$$IC(L') = \frac{\sum_j^{N_k} \overline{H}_j}{\sum_i^{N_l} \overline{H}_i}, \tag{4}$$

where N_k defines the number of streamlines in the reduced set. For an automatic streamline reduction procedure a threshold λ needs to be applied to $IC(L')$. The threshold guarantees that at least λ percent of the flow information remain in the subset.

4 Flow Field Distance

Applying techniques for featured streamline extraction and reduction, the quantification of the distance between two flow fields is rephrased to the quantification of the distance between sets of reduced featured streamline representations. Given that the distance $d_{m,n}$ between two corresponding streamlines m and n

of two representations A and B are given, the overall distance has been defined based on the accumulation of all $d_{m,n}$:

$$D(L'_A, L'_B) = \sum_{k}^{N_A} d_{m,n}, \tag{5}$$

with N_A defining the number of streamlines of representation A and l_m denoting the corresponding streamline to l_n. The calculation of the overall distance based on reduced streamline sets requires to detect corresponding streamlines between two representations, as well as the definition of an appropriate streamline wise distance measure $d_{m,n}$.

Identification of Corresponding Streamlines. The task of identifying corresponding streamlines is to assign a streamline of one flow field representation A to its comparable streamline of flow field B. This paper attempts the problem by assuming that corresponding streamlines are those which origin at the same or a near by seeding point. Given the initial point p^0_m of one streamline $l_m \in L^A$ with $u = 0$, the corresponding streamline in B is defined by the Euclidean closest point p^0_n over all streamline seeding positions in L^B, with $n \in [1, N_B]$. If the minimal distance is within a vicinity r the streamline l_n is said to be the corresponding streamline of l_m. The vicinity r is defined as the minimal distance between the seeding positions of neighboring streamlines considering all streamlines in A and B. If the minimal distance is not within the vicinity r a corresponding streamline to l_m is said to not exist.

Streamline Wise Distance. Given N_f different features and a related weight factor w_i, the streamline wise distance in its general form is denoted as:

$$d_{m,n} = \sum_{i}^{N_f} w_i \cdot d(P^m_i, P^n_i)$$

with $\sum_i^{N_f} w_i = 1.0$, and P^m_i, P^n_i defining the discrete feature distribution for feature i with respect to streamline l_m and the corresponding streamline l_n, respectively. In this paper four different probabilistic distance measures $d(P^m_i, P^n_i)$ have been considered to evaluate the distance between discrete feature distributions. The measures can be categorized into *bin-wise distances*: L^1 and χ^2 [9], and *cross-bin distances*: Earth Mover's Distance EMD [7] and the quadratic χ distance QC [4]. In the case that no corresponding streamline to l_m has been found, $d(P^m_i, P^n_i) = 0.0$.

5 Results

The flow distance quantification based on a reduced featured streamline representation has been evaluated given a flow field test data set, which covers velocity and feature data from the simulation of different polyhedral objects.

5.1 Experimental Setup

Various geometrical objects have been generated by modifying the baseline surface mesh of an icosahedron object. First, a subdivision algorithms[1] has been applied to smoothen the surface of the icosahedron, so that the object more and more morphs into a spherical shape, see Fig. 1. Second, objects with different level of subdivision have been rotated along the z-axis. The objects are denoted as ICO_{α}^{SL}, where $SL \in [0, 3]$ defines the subdivision level and $\alpha \in \{0, 10, \ldots, 90\}$ the rotation angle. In order to derive comparable flow fields, all objects have been constrained to a frontal area of $3m^2$. For the simulation of the flow around the

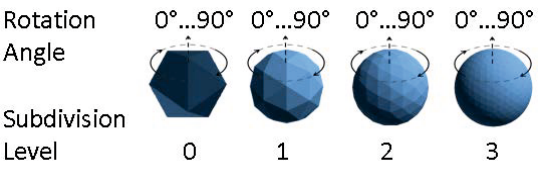


Fig. 1. Modified geometries for the test dataset have been generated by rotation and by subdividing the surface mesh of the icosahedron

polyhedral objects, each object has been positioned onto a ground plane with a distance of $48.0m$ from the inlet. In order to numerically solve the Navier stokes equations the vicinity around the object has been discretized into a volume mesh with around $350,000$ mesh cells using *snappyHexMesh*[2]. The flow has been modeled at a speed of $110km/h$. To solve the partial differential equations the simpleFOAM tool from *openFOAM*[3] has been applied. The velocity field linked to each object has been extracted from the time averaged flow. The initial set of streamlines has been uniformly seeded from a plain at the position $x = -48.0m$. The flow features along the individual streamlines have been calculated based on finite differences.

5.2 Entropy Based Streamline Reduction

In the first experiment a set of 450 streamlines has been generated for each of the 40 objects. By applying entropy based streamline reduction, first the relation between the ratio of the number of streamlines in the subsets and its related information content, (Eq. 4) have been under investigation. The streamline subsets are constraint to cover at least λ times the expected entropy of the entire streamline set. For different values of λ from 0.11 to 0.95 the related number of streamlines of the reduced subsets are depicted in Fig. 2 a). Each box denotes the statistics on the number of remaining streamlines over all different objects.

[1] `http://brainder.org/`, [2]
[2] `http://www.openfoam.org/docs/user/snappyHexMesh.php`
[3] `http://www.openfoam.com/`

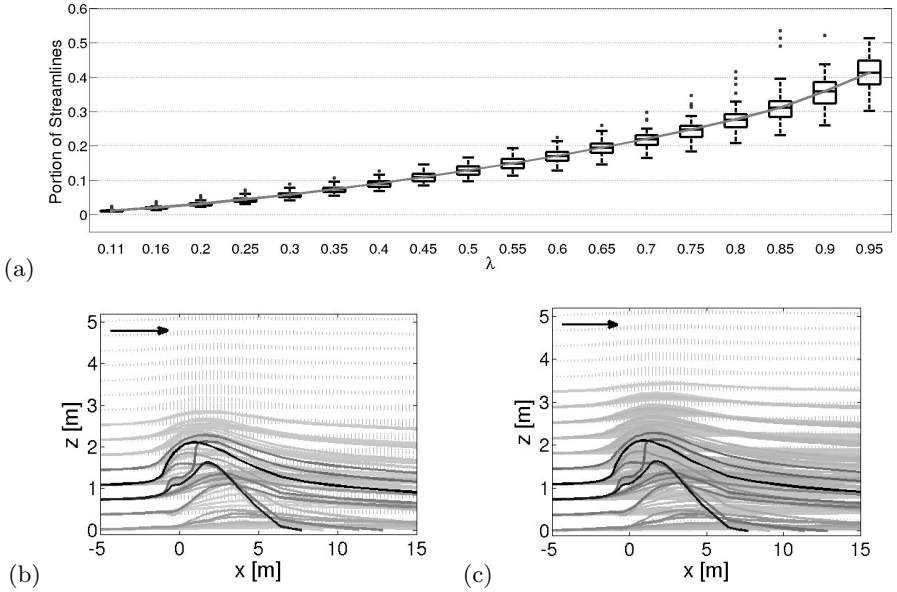


Fig. 2. a) Illustration of the portion of streamlines and the percentage of the related information kept after the streamline reduction step. Visualization of the statistics over all polyhedral objects in the test data set. The illustrations at the bottom show the reduced streamline sets regarding object Ico_0^0 with b) $\lambda = 0.50$ and c) $\lambda = 0.90$ (projected onto the x-z-axis).

As an example, with $\lambda = 0.95$ around 95% of the information about the flow features can be preserved by reducing the number of streamlines by more than 50%. The filtered 50% of the streamlines contribute only 5% to the expected entropy of the entire streamline data set. The results show that the suggested procedure for automatic streamline reduction can lead to a significant data reduction, reducing the memory consumption of the flow field representation.

Figs. 2 b) and c) depict two examples with $\lambda = 0.5$ and $\lambda = 0.9$, illustrating the reduced streamline data sets concerning object Ico_0^0. Solid lines show the most informative streamlines, with color proportional to the respective information content. Dotted lines highlight streamlines which have been removed from the dataset. It shows that the streamline ranking can simplify the visualization, by hiding all uninformative streamlines.

5.3 Flow Field Distance

In the following experiments it has been investigated how the distances between flow fields, which have been calculated based on the reduced streamlines, compare with the distances evaluated based on the complete streamline set. For reference, for each object three sets of uniform seeded streamlines with 450, 200 and 50 lines have been generated. The distances calculated based on these

three representations have been compared against the distances of three different compact streamline representations, with $\lambda \in \{0.9, 0.7, 0.5\}$. All in all, for each object six different representations have been generated. Related to each representation, the distances between the streamline data of all possible object combinations have been calculated according to Eq. 5, using either of the four distance measures (L^1, χ^2, EMD, $QC^B_{0.5}$).

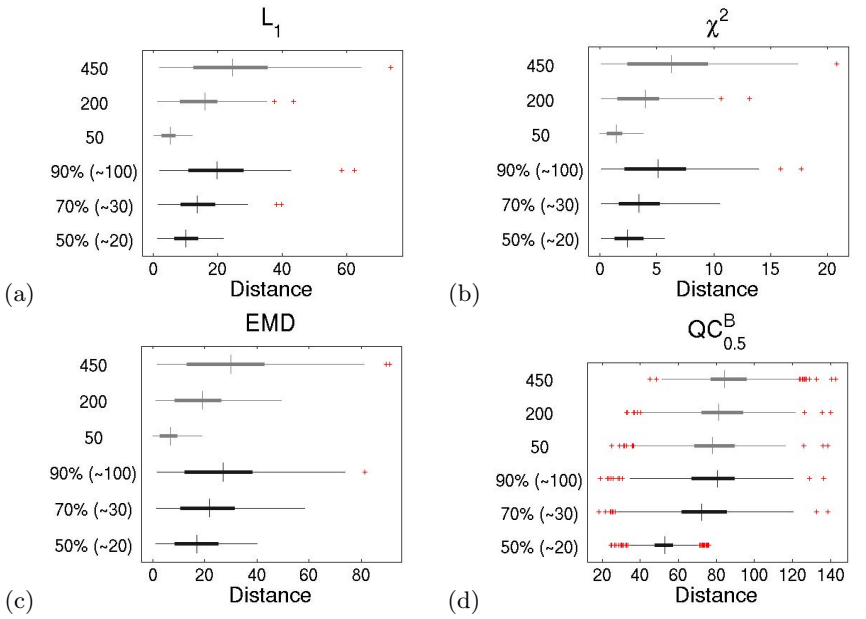


Fig. 3. Comparison of the flow field distances using either of the representation with fixed number of streamlines 450, 200 and 50 (gray) or using the compact streamline representation with $\lambda \in \{0.9, 0.7, 0.5\}$ (black).

The results of the experiments are depicted in Figs. 3 a) to d). Each bar denoting the distribution of the flow field distances of all object combinations, with all in all 1440 distance values. From all the different representation, it is assumed that the one with 450 uniform seeded streamlines most accurately represents the distances between the different flow fields. A decrease in the mean value over all flow field distances is expected to decrease the likelihood to correctly discriminate the flow fields. Considering the representations with uniform seeding, with 450, 200 and 50 streamlines, a reduction in the number of streamlines leads to a drop in the median distance value. Especially with only 50 streamlines a correct discrimination of the flow fields seems to be hardly possible. This effect can be similarly observed for all four distance measures.

In contrast, using the reduced streamline representation (reduced from 450 uniform seeded streamlines), the reference distance distribution can be almost

maintained even for low number of streamlines, independent of the distance measure used. As an example, consider Fig. 3 c), where the EMD has been used, it can be concluded that only around 100 streamlines are sufficient, with $\lambda = 0.5$, in order to maintain the distance distribution of the reference representation with 450 streamlines. This holds for almost any distance measure, except using the $QC^B_{0.5}$ distance. However, it can be concluded that around two third of the streamlines can be removed from the flow data, while maintaining almost the same distance values between flow fields.

6 Conclusion

In this paper a new attempt for deriving a reduced streamline wise flow field representation has been introduced, where streamlines are reduced based on the evaluation of its expected flow feature entropy. It has been demonstrated that the suggested procedure can lead to a significant reduction in memory consumption for storing the flow data, while maintaining the majority of the flow information. The quantification and analysis of the distance values between different flow fields, resulting from the simulation of different polyhedral objects, has turned out that the compact representation can almost maintain the same distances as a uniformly seeded streamline representation with a huge number of streamlines.

For consolidating the usability of such compact representations and the related technique for evaluating the streamline wise flow field distance, the same procedure will be applied to the simulation data of various passenger cars in the future. The resulting distances can be utilized then to visualize or automatically retrieve passenger car categories. Furthermore, the streamline wise distance measure needs to be extended to time-varying flow data.

Acknowledgement. The authors would like to thank Stefan Menzel and Sebastian Schmitt from the Honda Research Institute Europe GmbH for their valuable comments and advice.

References

1. Depardon, S., Lasserre, J., Brizzi, L., Bore, J.: Automated topology classification method for instantaneous velocity fields. Experiments in Fluids 42, 697–710 (2007)
2. Kenner, H.: Geodesic math and how to use it. Dome series. University of California (1976)
3. Laramee, R.S., Weiskopf, D., Schneider, J., Hauser, H.: Investigating swirl and tumble flow with a comparison of visualization techniques. In: Proceedings IEEE Visualization 2004, pp. 51–58. IEEE Computer Society (2004)
4. Pele, O., Werman, M.: The quadratic-chi histogram distance family. In: Danilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 749–762. Springer, Heidelberg (2010)

5. Post, F.H., Vrolijk, B., Hauser, H., Laramee, R.S., Doleisch, H.: Feature extraction and visualization of flow fields. In: Eurographics 2002 State-of-the-Art Reports, pp. 69–100 (2002)
6. Rath, M., Graening, L.: Modeling design and flow feature interactions for automotive synthesis. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 262–270. Springer, Heidelberg (2011)
7. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth movers distance as a metric for image retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)
8. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423 (1948)
9. Xu, L., Dinh, H.Q., Mordohai, P., Ramsay, T.: Detecting patterns in vector fields. In: AIAA Aerospace Sciences Meeting, Orlando, Florida, USA (2011)
10. Ye, X., Kao, D., Pang, A.: Strategy for seeding 3d streamlines. In: Visualization, VIS 2005, pp. 471–478. IEEE (2005)

MCGA: A Multiobjective Cellular Genetic Algorithm Based on a 3D Grid

Hu Zhang[1], Shenming Song[1,*], and Aimin Zhou[2]

[1] Center for Control Theory and Guidance Technology,
Harbin Institute of Technology, Harbin, 150001 China
jxzhanghu@126.com, songshenmin@hit.edu.cn
[2] Department of Computer Science and Technology,
East China Normal University, Shanghai, 200235 China
amzhou@cs.ecnu.edu.cn

Abstract. This paper proposes a new cellular multiobjective genetic algorithm based on a 3D grid structure. The basic idea is to organize the candidate solutions by a 3D grid, and the reproduction and replacement operators are based on the 3D grid. The proposed algorithm is compared with two 2D cellular multiobjective genetic algorithms on the DTLZ test suite, and the statistical results indicate that our approach performs better than the compared algorithms according to both the diversity and convergence metrics. Furthermore, our approach is computational more stable.

Keywords: multiobjective optimization, cellular genetic algorithm, 3D grid.

1 Introduction

Many engineering and real-world applications need to optimize multiple criteria simultaneously. This kind of problems are usually called multiobjective optimization problems (MOPs). Unlike scalar-objective optimization problems (SOPs), usually there does not exist a single solution which can optimize all the objectives at the same time. Instead, a set of tradeoff solutions among different objectives, called Pareto set (PS) in the decision space and Pareto front (PF) in the objective space, are of interests [1, 2].

Due to the nature of evolutionary algorithms (EAs), they have been widely applied to deal with MOPs. In the last decades, a variety of multiobjective evolutionary algorithms, such as NSGA-II [3], SPEA2 [4], PAES [5], cMOGA [6], aMOCell [7], MOEA/D [8], CellDE [9] and etc., have been proposed. Since EAs usually maintain a population of candidate solutions in the optimization process, they are able to approximate the PF (PS) of MOP in a single run. Further more, since the population size of MOEAs is limited, MOEAs aim to find an approximate set which is as diverse as possible and as close as possible to the PF in practice. Most of current MOEAs roughly fall into two categories: centralized algorithms and decentralized ones. For the

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 455–462, 2013.

former category, some examples include NSGA-II, SPEA2, and PAES. This kind of algorithms take the population as a whole and apply the reproduction and replacement operators in the whole population. For the latter category, also called structured algorithms, some examples are aMOCell4, CellDE, and MOEA/D. Unlike centralized algorithms, this kind of algorithms apply the reproduction and replacement operators in the local areas. In some cases, these decentralized algorithms provide a better sampling of the search space, which results in an improved numerical behavior with respect to an equivalent centralized algorithm [6,7,9].

This paper focuses on the cellular genetic algorithms (cGAs) in the second class. In a cGA [10], the population is conceptually set in a topological structure (also called population grid). According to the structure, several individuals in the close grid (neighbor grid) are assigned for each solution as its neighbors (neighborhood). Any given solution in cGAs only interacts with individuals belonging to its neighborhood. Many researches have been done for the scalar-objective cGAs (SOcGAs), but there have not existed plenty of achievements on cellular multiobjective genetic algorithms (cellMOGAs). As stated before, the typical cellMOGAs are: cMOGA, aMOCell4, CellDE etc., and these cellMOGAs have been proved that there are outstanding performance in some conditions.

From the statement above, we can find that, the grid is an important characteristic of the cGAs, it can be 1, 2, 3 dimensions [10], and the SOcGAs with these three kinds of grid structrures have been investigated largely. However, in current study on the cellMOGAs, the population individuals are always arranged in a two dimensional (2D) grid. Thus, we plan to introduce the three dimensional (3D) structure into the cellMOGA, and design a 3D cellular multiobjective genetic algorithm (MCGA). As discussed in the case of scalar-objective optimization [11-13], due to the shorter radius and denser neighborhood that result from the vertical expansion of cells, the 3D structure is able to facilitate faster spreading of solutions, and the higher cellular dimensions can improve the algorithm performance. We hope that the specialty of 3D cellular structure could also play a positive role in the cellMOGAs, and the proposed MCGA will have a better optimization performance.

2 A 3D Multiobjective Cellular Genetic Algorithm

This section presents a new multiobjective cellular genetic algorithm based on a 3D population grid. Before introducing our new approach, the canonical cellular genetic algorithm is summarized firstly.

2.1 A Canonical Asynchronous Cellular Genetic Algorithm

The pseudocode of cGA is included in Algorithm 1. In an asynchronous cGA, the population is usually structured in a regular grid of d dimensions ($d = 1,2,3$), and a neighborhood is defined on it. The algorithm iteratively does the genetic operations on each individual in the grid (line 3). An individual may only interact with individuals belonging to its neighborhood (line 4), and its parents are chosen among its neighbors (line 5) with a given criterion. Crossover and mutation operators are applied to the individuals in lines 6 and 7, with probabilities Pc and Pm, respectively.

Afterwards, the algorithm computes the fitness value of the new offspring individual (or individuals) (line 8), and inserts it (or one of them) into the equivalent place of the current individual in the population (line 9) following a given replacement policy. This loop is repeated until a termination condition is met (line 2).

Algorithm 1. Pseudo-code of a Canonical Asynchronous cGA

```
1.    proc Steps_Up(cga) //Algorithm parameters in 'cga'
2.    while not Termination_Condition() do
3.    for individual ← 1 to cga.popSize do
4.      neighbors←Get Neighborhood(cga,position(individual));
5.      parents←Selection(neighbors);
6.      offspring←Recombination(cga.Pc,parents);
7.      offspring←Mutation(cga.Pm,offspring);
8.      Evaluate_Fitness(offspring);
9.      Replacement(position(individual),offspring,cga);
10.   end for
11.   end while
12.   end proc Steps_Up;
```

2.2 Extending cGA for Multiobjective Optimization

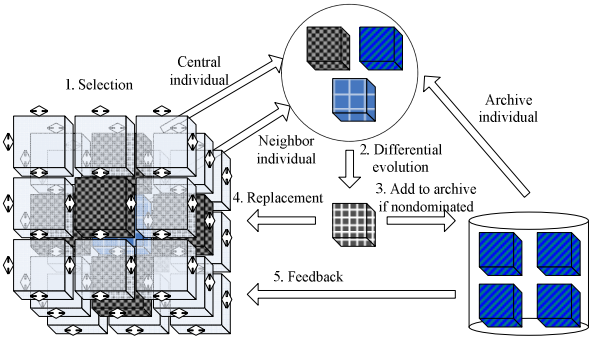


Fig. 1. Breeding loop of MCGA

Based on the previously cGA model, a multiobjective cellular genetic algorithm (MCGA) is proposed in this section. Unlike other cellular algorithms for mutiobjective optimization, MCGA uses a 3D grid to organize the population. Algorithm 2 shows the pseudo code of MCGA. The Pareto front is just an additional population (the external archive) composed of the non-dominated solutions found. The archive has a maximum size and, therefore, the insertion of solutions in the Pareto front has to be carefully managed to obtain a diverse set. Hence, a density estimator is needed to remove solutions from the archive when it becomes full.

A. Breeding Loop: MCGA starts by creating an empty Pareto front (line 2 in Algorithm 2). Individuals are arranged in a 3-dimensional toroidal grid, and the DE operations are successively applied to them (lines 9) until the termination condition is met (line 3). The breeding loop of MCGA is illustrated in Figure 1. Hence, for each

individual, three parents are selected, one is from the neighborhood, the other comes from the archive, and the last one is the current individual itself. After selection, the algorithm proceeds a DE operation to obtain a new offspring, and evaluates the resulting individual; then the algorithm decides whether the new offspring replaces the current one (line 11). The next step (line 12) is to insert the offspring into the external archive, if appropriate. Finally, after each generation, a feedback procedure is invoked to replace a number of randomly chosen individuals by a number of solutions from the archive (line 14).

B. Replacement: In this algorithm, the resulting offspring replaces the individual at the current position if the former is better than the later. The approach of defining the concept of "better individual" is to replace the current individual if it is dominated by the offspring or both the two are non-dominated and the current individual has the worst crowding distance (as defined in NSGA-II) in a population composed of the neighborhood plus the offspring.

C. Archive Maintenance: In order to insert offspring individual into the Pareto front (line 12 in Algorithm 2), following mechanism is adopted: the dominance relationships between offspring individual and the solutions in the archive are judged first. After that, the archive solutions dominated by offspring individuals are all removed, and then if the offspring is dominated by any remaining archive solutions, the offspring individual will not be added; otherwise, it will become one of the solutions in the archive. If the external archive is full, a fitness value will be assigned to each solution in the archive (including offspring individual) through the method proposed in SPEA2 [4], and then the solutions with the maximum value will be removed.

Algorithm 2. Pseudo-code of MCGA

```
1.   proc Steps Up(MCGA) //Algorithm parameters in 'MCGA'
2.   ParetoFront = Create Front() //Creates an empty Pareto front
3.   while not TerminationCondition() do
4.   for individual ← 1 to MCGA.popSize do
5.     neighbors←Get Neighborhood(MCGA,position(individual));
6.     Parent1←Binary Tournament Selection(neighbors);
7.     Parent2←Random Selection(ParetoFront);
8.     Parent3←individual;
9.     offspring←Differential Evolution(parent1,parent2,parent3,CR,F);
10.    Evaluate Fitness(offspring);
11.    Replacement(position(individual),offspring, MCGA);
12.    Insertion(ParetoFront,offspring);
13.  end for
14.  MCGA.pop←Feedback(MCGA,ParetoFront);
15.   end while
16.   end proc Steps Up;
```

3 Experiments

To evaluate the MCGA, we apply the MCGA to the DTLZ suite [14], and the Inverted Generational Distance (*IGD*) metric [17] is used to assess the algorithm per-

formance. We also compare MCGA with two typical and excellent cellMOGAs: aMOCell4 [7] and CellDE [9]. The details of test problems and comparison algorithms could be found in Ref. [14,15,9].

3.1 Settings

The terminal conditions of three algorithms are 25000 maximum problem evaluations. The sizes of external archives in three algorithms are 125. 25 randomly selected individuals make up the feedback population. The statistical results are based on 30 independent runs.

In aMOCell4, the population size is 121, and the individuals are arranged in a 11×11 square grid. The neighbor structure is NEWS (i.e. north, east, west, south). The basic genetic operators are binary tournament selection, simulated binary crossover and polynomial mutation. The crossover and mutation probability is $P_c = 0.9$ and $P_m = 1/n$, and the distribution index in crossover and mutation is $\eta = 20$.

In CellDE, the population size, grid shape and neighbor structure are the same with those in aMOCell4. The control parameters in DE are $F = 0.5$ and $CR = 0.1$.

In MCGA, the population size is 125, and the grid is a $5\times5\times5$ cube. The neighbor grid is 3D NEWS (i.e. horizontal north and south, vertical north and south, east, and west). The control parameters of DE are the same with those in CellDE.

3.2 Results and Analysis

Table 1 includes the mean and standard deviation of the IGD values when aMOCell4, CellDE and MCGA compute each DTLZ problem for 30 times independently. The data with the dark grey background are the best values, and the ones with light grey background are the suboptimum values. It is easy to observed that, among the 7 problems, MCGA obtains 4 optimal and 1 suboptimum values, CellDE gets 2 optimal and 5 suboptimum values, and aMOCell4 just has 1 optimal and 1 suboptimum values. Since the IGD indicator is able to response the convergence and diversity of the algorithms simultaneously, the data in Table 1 indicates that, MCGA has the best performance for solving the DTLZ benchmark; CellDE owns a better effect, and the aMOCell4 is the worst.

Table 1. The statistical results of IGD values obtained by aMOCell4, CellDE and MCGA

	aMOCell4		CellDE		MCGA	
	mean	std.	mean	std.	mean	std.
DTLZ1	0.325224	0.297632	0.278245	0.271817	0.417448	0.434799
DTLZ2	0.060037	0.002837	0.050477	0.001106	0.050662	0.001098
DTLZ3	10.116994	5.033764	9.568282	8.598339	7.799721	4.418446
DTLZ4	0.098114	0.110569	0.042147	0.003636	0.042096	0.004570
DTLZ5	0.003237	0.000064	0.003914	0.000193	0.004110	0.000196
DTLZ6	2.356555	0.204823	0.003575	0.000170	0.003569	0.000155
DTLZ7	0.285042	0.262678	0.161212	0.175893	0.156526	0.186900

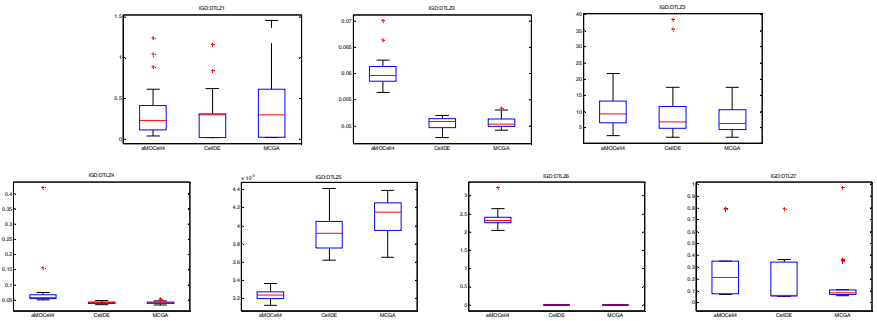
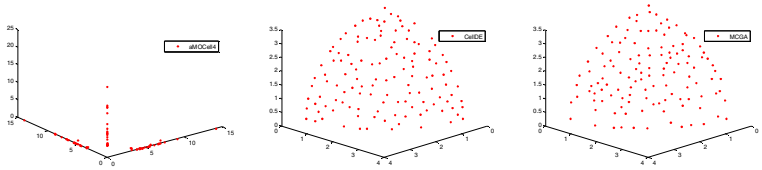


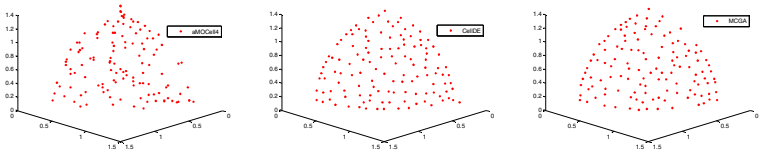
Fig. 2. The box plots of the *IGD* values obtained by aMOCell4, CellDE and MCGA

To describe the statistical results of *IGD* values directly, we plot the box plots for each problem in Figure 2. In the box plot, the rectangle represents the interquartile range of the 30 *IGD* values, and its physical significance is the distribution of the *IGD* values. If the rectangle is much shorter, it means that the values distributed more densely, and the algorithm has a more stable performance. The transverse line in the rectangle denotes the median of the *IGD* values, obviously, a smaller median is desirable. From the figures above, it can be noted that, compared with other two algorithms, MCGA has 5 shorter rectangles (i.e. DTLZ2-4, DTLZ6-7), and it obtains 4 smaller median values (i.e. DTLZ2-4, DTLZ6), which implies that MCGA is able to reach a robust effect and gets the solutions with better convergence and diversity while solving DTLZ problems.

In order to show the results visually, we plot the distributions of the best PFs (i.e. it has the minimum IGD value) obtained by aMOCell4, CellDE and MCGA for problem DTLZ3-4, 6-7 in Figure 3. Obviously, it is easy to known that aMOCell4 has the worst distributions. With regard to CellDE and MCGA, it is hard for us to discover the differences. However, if we observe more carefully, we are still able to find that MCGA are slightly better than CellDE (for example: the area of bottom right corner in Figure 3(b)).

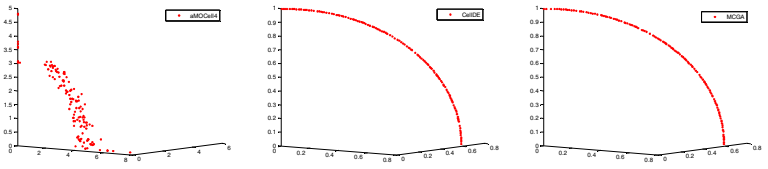


(a) DTLZ3

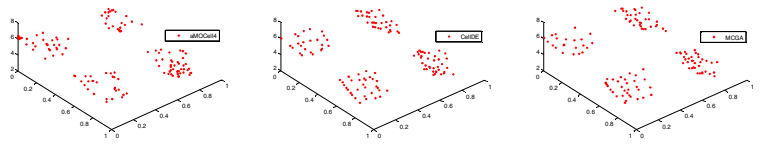


(b) DTLZ4

Fig. 3. The best PF approximations obtained by aMOCell4, CellDE and MCGA



(c) DTLZ6



(d) DTLZ7

Fig. 3. (*continued*)

4 Conclusions and Future Work

In this paper, we have proposed a new multiobjective cellular genetic algorithm, called MCGA, for continuous mutiobjective optimization. Comparing with related multiobjective cellular algorithms, our major contribution is that a 3D grid is utilized to organize the population. In MCGA, three parents are selected from the neighborhood defined in the 3D grid, the archive, and the central individual itself, the DE operators are adopted to generate new trial solutions, and the trial solutions are then applied to update the neighborhood and archive. MCGA was compared with two related algorithms on the DTLZ test suite, and the experimental results suggested that MCGA showed the best performance according to both the diversity and convergence metrics, and it was computational more stable.

Due to the page limitation of here, only the DTLZ benchmark was tested. The problems with more objectives and complicated PF (or PS) shapes should be used to evaluate the performance of MCGA. Besides, some interesting mechanisms adopted in 2D cGAs, such as: adaptive, two neighborhood and so on, can be brought into MCGAs to design more effective MOGAs.

Acknowledgment. This work has been funded by National Natural Science of China (Grant No. 61174037, 61273313, and 51275274).

References

1. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
2. Miettinen, K.M.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)

3. Deb, K., Pratap, A., Agarwal, A., et al.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Kim, M., Hiroyasu, T., Miki, M., Watanabe, S.: SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 742–751. Springer, Heidelberg (2004)
5. Knowles, J., Corne, D.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation. In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC 1999), pp. 98–105. IEEE (1999)
6. Alba, E., Dorronsoro, B., Luna, F., et al.: A Cellular Multi-objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs. Computer Communications 30(4), 685–697 (2007)
7. Nebro, A.J., Durillo, J.J., Luna, F., Dorronsoro, B., Alba, E.: Design Issues in a Multiobjective Cellular Genetic Algorithm. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 126–140. Springer, Heidelberg (2007)
8. Zhang, Q.F., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transation on Evolutionary Computation 11(6), 712–731 (2007)
9. Durillo, J.J., Nebro, A.J., Luna, F., Alba, E.: Solving Three-Objective Optimization Oroblems Using a New Hybrid Cellular Genetic Algorithm. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N., et al. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 661–670. Springer, Heidelberg (2008)
10. Abla, E., Dorronsoro, B.: Cellular Genetic Algorithms. Springer, Berlin (2008)
11. Al-Naqi, A., Erdogan, A.T., Arslan, T.: Balancing Exploration and Exploitation in an Adaptive Three-dimensional Cellular Genetic Algorithm via a Probabilistic Selection operator. In: Proceedings of 2010 NASA/ESA Conference on Adaptive Hardware and Systems, pp. 258–264. IEEE Computer Society (2010)
12. Al-Naqi, A., Erdogan, A.T., Arslan, T.: Fault Tolerance Through Automatic Cell Isolation Using Three-dimensional Cellular Genetic Algorithms. In: Proceedings of 2010 IEEE Congress on Evolutionary Computation (2010)
13. Al-Naqi, A., Erdogan, A.T., Arslan, T.: Fault Tolerant Three-dimensional Cellular Genetic Algorithms with Adaptive Migration Achemes. In: Proceedings of 2011 NASA/ESA Conference on Adaptive Hardware and Systems, pp. 352–359. IEEE Computer Society (2011)
14. Deb, K., Thiele, L., Laumanns, M., et al.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Proceedings of the Evolutionary Multiobjective Optimization, pp. 105–145. Springer, Heidelberg (2005)
15. Durillo, B.J.J.: Metaheuristics for Multi-objective Optimization: Design, Analysis, and Applications. University of Malaga, Spain (2011)

Multi-Objective Particle Swarm Optimization Algorithm Based on Population Decomposition

Yuan Zhao and Hai-Lin Liu*

School of Applied Mathematics
Guangdong University of Technology
Guangzhou, China
yuanyuan9995@126.com,
hlliu@gdut.edu.cn

Abstract. In this paper, a novel multi-objective particle swarm optimization algorithm is proposed based on decomposing the objective space into a number of subregions and optimizing them simultaneously. The subregion strategy has two very desirable properties with regard to multi-objective optimization. One advantage is that the local best in the subregion can effectively guide the particles to Pareto front combining with global best. The other is that it has a better performance on the convergence and diversity of solutions. Additionally, this paper applies min-max strategy with determined weight as fitness functions to multi-objective particle swarm optimization, and there is no additional clustering or niching technique needed. In order to demonstrate the performance of the algorithm, it is compared with MOPSO and DMS-MO-PSO. The results indicate that proposed algorithm is efficient.

Keywords: Particle swarm optimization, multi-objective, population decomposition.

1 Introduction

Particle swarm optimization algorithm (PSO) has been found to be an efficient optimization method for single objective optimization[1][2]. Recently, it shows promising results for solving multi-objective optimization problems[6][7][4]. In most multi-objective particle swarm optimization algorithms, the fitness assignment scheme plays a key role. Usually, the selection strategies can be classified as Pareto-domination[5][6][7][10] or aggregating objectives[4][8]. X.Li[5] proposed a multi-objective particle swarm optimization based on non-dominated sorting(NSPSO). Instead of a single comparison between a particle's personal best solution and its offspring, NSPSO compares all particles' personal best solutions with all of offsprings. It uses parameter-free niching method to maintain diversity. In[6], an archive set was applied to store the non-dominated solutions found so far, and a mutation operator is used to ensure the population diversity.

A multi-objective particle swarm optimization based on MOEA/D[3] was proposed in[8]. This paper adopts a decomposition approach to select global best

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 463–470, 2013.

solutions through the boundary intersection. It also uses a mechanism to reinitialize the particles based on their age. The fitness function with the max-min strategy was proposed by Li[4]. This paper adopts a similar method as NSPSO[5]. It uses the max-min fitness to rank individuals in the population (rather than the non-dominated sorting procedure), and it doesn't use niching method.

In the multi-objective particle swarm optimization(MOPSO) algorithms above, a global best solution is selected to guide other solutions to the Pareto front. Since the selected global best solution has greater impact on those particles near it than those far away from it, the effect of global search strategy is not very good. However, in recent years, the subpopulation concept is incorporated with multi-objective particle swarm optimization(MOPSO)[7][9][10]. Coello et al.[7] proposed an improved MOPSO version (AMOPSO), in which a clustering technique is used to divide the population into several swarms aiming to maintain a better distribution of solutions. A multiple swarms algorithm was proposed in[9]. Several components, such as cell-based rank density estimation, population growing and declining strategies, and adaptive local search, are designed to improve the algorithmic performance. A novel pbest and lbest updating criteria was proposed by J.J.Liang[10]. This paper divides the population into several swarms randomly. The swarms are regrouped randomly after definite generations. The external archive set is also divided into many subsets according to one objective, its number is same as subswarms'. The local best position of a swarm is randomly selected from anyone of these archive subsets. It can maintain the diversity of the population.

In order to get a set of solutions with good convergence and spread on the Pareto front, a novel multi-objective particle swarm optimization algorithm based on population decomposition(MOPSO-PD) is proposed. It divides the objective space into a number of subregions and optimizes the objectives respectively. In each generation, a local archive set of each subregion and a global archive set are set up. So there are three factors affecting the speed of particles in each subregion: personal best solution, local best solution and global best solution. The proposed algorithm makes full use of these three best solutions when updating speed. They can effectively guide those particles to approach Pareto front in each generation, and hence a better convergent and well-distributed performance follows. Then the max-min selection strategy is used with prior determined weights to form fitness functions. These approaches can reduce the computational complexity and maintain diversity and uniformity of the population.

The rest of this paper is organized as follows: Section 2 introduces MOPSO-PD, Section 3 gives the experimental results. Section 4 concludes this paper.

2 MOPSO-PD

Normally, a multi-objective optimization problem can be described as follows:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), \ldots, f_m(x)) \\ &\text{subject to} \quad x \in \prod_{i=1}^{D} [a_i, b_i] \subset R^D \end{aligned} \tag{1}$$

where x is the decision variable, $\prod_{i=1}^{D}[a_i, b_i] \subset R^D$ is the decision space, $f_i(x)$ is a objective function, $i = 1, 2, \ldots, m$. Since these objectives are contradictory or conflicting, there is no solution to minimize all objectives simultaneously. So we need to present a fitness assignment method to choose better individuals. The min-max strategy is used in this paper.

2.1 Decomposition of Swarm

Two major goals of multi-objective optimization algorithm are to obtain a set of non-dominated solutions as close as possible to the Pareto front, and to maintain well-distributed solutions along the Pareto front. In order to get them, this paper proposes a novel MOPSO by decomposing the objective space into S subregions. Firstly, we give N points which are uniformly distributed on the straight line (or equilateral triangle in three-dimensional space) denoted as the weight vectors. Consequently, we have N weight vectors $W^1, W^2, \ldots W^N$ where $W^i = (w_1^i, w_2^i, \ldots w_m^i)$. We denote their corresponding points as $A_1, A_2, \ldots A_N$. Similarly, we can design S weight vectors ($S < N$), corresponding evenly distributed points on the straight line (or equilateral triangle in three-dimensional space) denoted as $T_1, T_2, \ldots T_s$. Each vector represents the center of a sub-region. By calculating the distance between the weight vector and the center vector, all weight vectors are assigned to corresponding subregions according to the shortest distance. We use min-max strategy with determined weight as fitness functions. The ith fitness function is defined as follows:

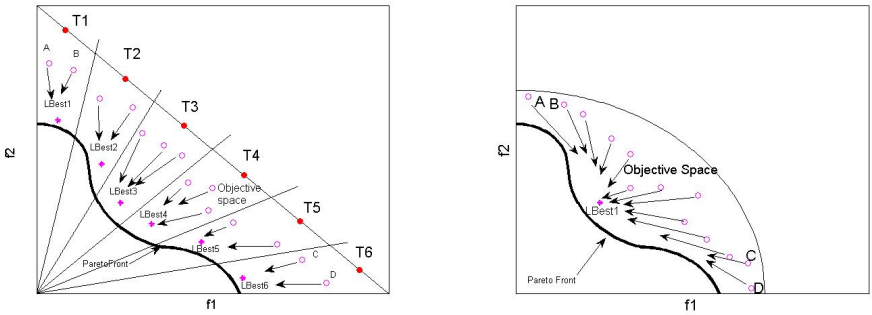
$$F_i(x) = \max_{1 \le j \le m} w_j^i f_j(x) \tag{2}$$

where w_j^i is jth dimension of ith weight vector W. $f_j(x)$ is the jth objective function. The weight vector in each subregion is used to select corresponding particle to subregion in initialization.

When $m = 2$, as shown in Fig.1(a), suppose $S = 6$, we give six points which are uniformly distributed on the straight line. Each vector represents the center of a sub-region. Then the population are divided into 6 subregions. A, B are allocated to the first region, and C, D are allocated to the 6th region accordingly . A and B are guided to the Pareto front by lbest1, while C and D are guided by lbest2. The convergence speed of algorithm is much faster, we can also get well-distributed Pareto solutions. However, in Fig.1(b), particles are guided by pbest and gbest. The guidance of gbest is not obvious to boundary particles A, B, C, D in aspect of convergence. The convergence speed of algorithm is slower.

2.2 Particle Update Mechanism

In particle swarm optimization algorithm, the important roles are pbest and gbest (lbest) which are used to record the historical information. Then how to update them becomes a problem. Here, we use min-max strategy with determined weight as fitness functions to update them as follows:



(a) (b)

Fig. 1. (a)Subregion method based on the objective function space; (b)Search in the global objective function space

(1) If $\min_{x \in X} F_i(x) = F_i(x_i^t)$ then $pbest^t[i] = x_i^t$.
(2) If $\min_{x \in X} F_i(x) = F(pbest^{(t-1)}[i])$ then $pbest^t[i] = pbest^{(t-1)}[i]$.
(3) If $\min_{x \in X} F_i(x) = F(pbest^{(t-1)}[i]) = F_i(x_i^t)$ then select one from them as the pbest of the tth generation randomly.

Update the archive set. In each subregion, we put all the personal best solutions into local archive set. The lbest is selected from the set randomly. A global archive set is also added to the proposed algorithm. In each generation, it generates N new positions. Firstly, we put them into the global archive set. There are $2N$ individuals Together with the previous generation of non-dominated solutions. Then we use N weights to select N non-dominated solutions and put them into global archive set. The gbest can be selected at random from the global archive set. Each swarm is evolved by updating both the velocity v^{t+1} and the position x^{t+1} of each particle according to the following equations:

$$v_{id}^{(t+1)} = wv_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{ld}^t - x_{id}^t) + c_3 r_3 (p_{gd}^t - x_{id}^t) \quad (3)$$

$$x_{id}^{(t+1)} = x_{id}^t + v_{id}^{(t+1)} \quad (4)$$

where r1 and r2 are uniform random numbers in $(0, 1)$, $c_1, c_2 \geq 0$ are learning factors and $w \geq 0$ is inertia weight. It determines how much of the previous velocity inherited. $1 \leq d \leq D$, $1 \leq i \leq N$. p_{id}^t, p_{ld}^t, p_{gd}^t are personal best position, local best position and global best solution. They guide particles to Pareto front together. Lbest is selected from the local archive set. It can maintain a set of well-distributed solutions. Gbest is selected from global archive set. It can accelerate the convergence of the algorithm. At the same time we should consider the limits of position and velocity. If $v_{id}^t \geq v_{max}^t$, then let $v_{id}^t = v_{max}^t$. If $v_{id}^t \leq v_{min}^t$, then let $v_{id}^t = v_{min}^t$. If $x_{id}^t \geq U_d^t$, then let $x_{id}^t = U_d^t$. If $x_{id}^t \geq L_d^t$, then let $x_{id}^t = L_d^t$.

2.3 The Main Framework Of MOPSO-PD

The main framework of MOPSO-PD is as follows:

Algorithm 1. The Main Framework Of MOPSO-PD.

Step 1. Initialization:

Step 1.1. Set the population size N, and the number of class S, current evolution generation t, the maximum number of evolution generation T.

Step 1.2. Design the weighted vectors $w_1, w_2, \ldots, w_N$, classify all the weighted vectors into S classes, and set the representative point T_s in each category.

Step 1.3. Initialize particle position, velocity, pbest, local archive and global archive. $pop_0[i]$, $vel_0[i] = 0$, $pbest_0[i] = pop_0[i]$, $i = 1, 2, \ldots, N$.

Step 1.4. Classify N new individuals into each category in accordance with fitness functions with determined weight.

Step 2. Update:

Step 2.1. Evaluate the population, update the X_i and V_i as the formula (3)(4).

Step 2.2. Using the weight to update the pbest for each individuals and the local archive set.

Step 2.3. Update the archive set: using the N weights to select N non-dominated individuals from the N new individuals and old archives.

Step 3. Stopping:

Step 3.1. If $t \leq T$, go to step 2.1; otherwise,stop and output all the non-dominated solutions from global archive set.

3 Experimental Results

3.1 Test Instances

In order to demonstrate the performance of the proposed algorithm, we use bi-objectives ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 instances and two 3-objectives DTLZ1, DTLZ2 test instances[3]. All of these test instances have no constraint. Their search space is $[0, 1]^D$, where D is the dimension of decision space. $n = 30$ for ZDT1, ZDT2, ZDT3 instances and $n = 10$ for others.

3.2 Experimental Settings

In this experiment, all the algorithms stop after 20000 function evaluations, and the maximum generations is round $20000/N$. The MOPSO is given a population size of 100, a repository size of 100 and 30 divisions for the adaptive grid with mutation as presented in [6]. DMS-MO-PSO is given a population size of 100, witch has 20 sub-swarms with 5 particles in each sub-swarm, and its regroup period $R = 5$. MOPSO-PD uses the following parameter values: Population size is 100. The objective space is divided into 20 sub-regions, global archive size $N_{max} = 100$. In our experiments, the IGD-metric[11] and H-metric[12] are used to measure the quality of a solution set P.

3.3 Computation Simulation

We have compared MOPSO-PD with DMS-MO-PSO, MOPSO on multi-objective optimization problem. All algorithms are run 30 times, the mean and best of

Table 1. The mean and best of IGD-metric values

IGD-metric	MOPSO-PD		DMS-MO-PSO		MOPSO	
Instance	Mean	Best	Mean	Best	Mean	Best
ZDT1	0.0053	**0.0039**	**0.0048**	0.0042	0.1248	0.0099
ZDT2	**0.2782**	**0.0038**	0.5087	0.6097	0.5530	0.1583
ZDT3	0.0099	0.0085	**0.0063**	**0.0052**	0.5524	0.4246
ZDT4	**0.0040**	**0.0039**	0.1571	0.0055	0.0775	0.0258
ZDT6	**0.0459**	**0.0436**	0.0569	0.0439	0.0534	0.0465
DTLZ1	**0.0558**	**0.0547**	0.2027	0.2905	0.3512	0.2084
DTLZ2	**0.0717**	**0.0687**	0.1958	0.1627	0.4079	0.3315

Table 2. The mean and best of H-metric values

H-metric	MOPSO-PD		DMS-MO-PSO		MOPSO	
Instance	Mean	Best	Mean	Best	Mean	Best
ZDT1	**0.6609**	**0.6616**	0.5921	0.6631	0.5010	0.6493
ZDT2	**0.1771**	**0.3285**	0.2283	0.2604	0.0246	0.2033
ZDT3	**0.7743**	**0.7761**	0.6783	0.7067	0.3882	0.5368
ZDT4	**0.6611**	**0.6613**	0.5633	0.6405	0.6184	0.6468
ZDT6	**0.3222**	**0.3225**	0.3076	0.3152	0.3106	0.3195
DTLZ1	**0.7343**	**0.7413**	0.5737	0.6052	0.3921	0.5067
DTLZ2	**0.3772**	**0.3823**	0.2242	0.2462	0.1539	0.1923

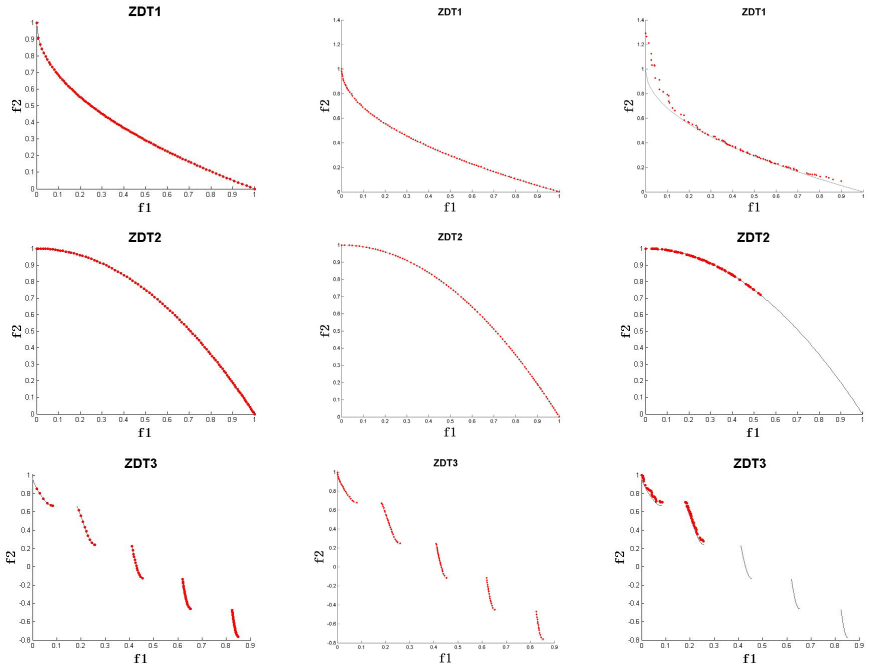


Fig. 2. Plot of the nondominate fron with the median IGD-metric value found by MOPSO-PD(the left panel),DMS-MO-PSO(the middle panel) and MOPSO(the right panel)

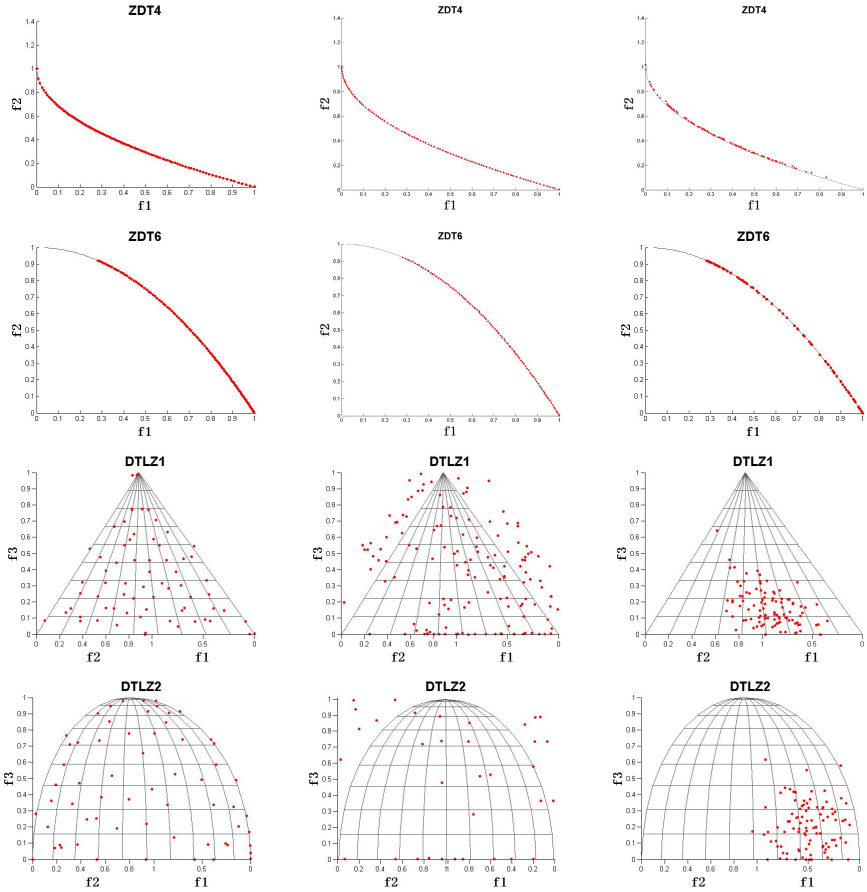


Fig. 2. (*continued*)

IGD-metric value of MOPSO-PD, DMS-MO-PSO and MOPSO are shown in table 1, bolds indicate best value obtained of the three algorithms. From table 1, we can see that for convex function ZDT1 and ZDT3, DMS-MO-PSO is better than our algorithm MOPSO-PD. However, to other test functions, proposed algorithm MOPSO-PD is better. The mean and best of H-metric value are shown in table two, bolds indicates best value obtained of the three algorithms. Obviously, in table 2 our algorithm is much better than other algorithms.

In the following Fig.2, the spread of DMS-MO-PSO and MOPSO-PD is better than MOPSO for 3-objectives DTLZ1, DTLZ2 test instances. It is clear that the convergence of MOPSO-PD is better than MOPSO and DMS-MO-PSO for some bi-objectives instances ZDT2, ZDT4, ZDT6. It is relatively simple for bi-objectives instances. However, the MOPSO-PD is perfect in 3-objectives instances.

4 Conclusion

This paper has proposed a novel multi-objective particle swarm optimization algorithm based on sub-regional search, in which the objective space is divided into several subregions and then optimized each sub-regional. Compared with those algorithms above, our algorithm has less computational complexity. Whats more, we uses min-max strategy with determined weight as fitness functions. It could effectively avoid falling into local optimum, and obtain a uniform distribution solutions. The numerical results indicate that proposed algorithm is efficient.

Acknowledgment. This work was supported in part by the Natural Science Foundation of Guangdong Province (S2012010008813), Programme of Science and Technology of Guangdong Province (2012B091100033), Programme of Science and Technology of the Department of Education of Guangdong Province (2012KJCX0042), and Zhongshan Programme of science and technology (20114A223).

References

1. Eberhart, R.C., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proc. MHS 1995, Nagoya, Japan, pp. 39–43 (1995)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
3. Li, H., Zhang, Q.: MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11, 712–731 (2007)
4. Li, X.: Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 117–128. Springer, Heidelberg (2004)
5. Li, X.: A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 37–48. Springer, Heidelberg (2003)
6. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling Multiple Objectives with Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation 8(3), 256–279 (2004)
7. Toscano Pulido, G., Coello Coello, C.A.: Using Clustering Techniques to Improve the Performance of a Multi-objective Particle Swarm Optimizer. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 225–237. Springer, Heidelberg (2004)
8. Martnez, S.Z., Coello Coello, C.A.: A Multi-objective Particle Swarm Optimizer Based on Decomposition. In: 13th Annual Conference on GECCO, pp. 69–76 (2011)
9. Yen, G.G., Wen, F.L.: Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization. IEEE Transction on Systerm Man, and Cybernetics Part A: Systems and Humans 39(4), 890–911 (2009)
10. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimization for Multi-Objective Optimization Problems. In: WCCI 2012, pp. 10–15 (June 2012)
11. Liu, H.-L., Li, X.: The Multiobjective Evolutionary Algorithm based on Determined weight and Sub-regional Search. In: CEC 2009, pp. 1928–1934 (May 2009)
12. Liu, H.-L., Gu, F., Zhang, Q.: Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems. In: IEEE Trans. Evol. Comput. (in press, 2013)

An Effective Ant Colony Approach for Scheduling Parallel Batch-Processing Machines

Rui Xu[1,⋆], Huaping Chen[1], and Hao Shao[2]

[1] School of Computer Science and Technology,
University of Science and Technology of China,
230026 Hefei, China
[2] School of WTO Research & Education,
Shanghai University of International Business and Economics,
200336 Shanghai, China

Abstract. This paper investigates the scheduling problem of parallel batch-processing machines which involves the constraints of non-identical job sizes, unequal release times, and batch dependent processing times for minimizing makespan. We proposed an Ant Colony Optimization (ACO) algorithm to solve the problem. of the ACO algorithm. In order to utilize the available information and obtain a tradeoff between exploitation and exploration, a novel construction policy and an efficient candidate list strategy were introduced during the process of solution construction of the ACO algorithm. Through extensive computational experiments, the effectiveness of the proposed algorithm was validated on different test problems. The results demonstrated that the proposed ACO algorithm had a superior performance compared to other benchmark algorithms.

Keywords: Scheduling, Parallel batch-processing machines, Makespan, Ant colony optimization.

1 Introduction

In this paper, we address the problem of scheduling parallel batch-processing machines (PBPMs) with non-identical job sizes and dynamic job arrivals. PBPMs are usually required in real-world production systems in order to prevent the

⋆ This work was supported in part by the National Natural Science Foundation of China (No.71171184/61175065/71201151), the Funds for Creative Research Group of China (No. 70821001), the NSFC major program (No.71090401/71090400), the China Postdoctoral Science Foundation (No.2011M501067/2013T60628), the Fundamental Research Funds for the Central Universities (No.WK0110000032), the Program for New Century Excellent Talents in University (No.NCET-12-0512), the Science and Technological Fund of Anhui Province for Outstanding Youth (No.1108085J16), the Humanity and Social Science Youth foundation of Ministry of Education of China (No.13YJC630126), and the Cai Yuanpei Program (No.27927VE).

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 471–478, 2013.

system form being blocked by the unavailability of a single batch-processing machine (BPM). Non-identical job size indicates that each job has arbitrary size, which doesn't exceed the maximum machine capacity. Dynamic job arrival means that jobs are not always available at the beginning of the scheduling horizon. The makespan (C_{max}) objective is adopted to evaluate the completion time performance. The target problem can be denoted by $P_m|r_i, s_i, p\text{-}batch|C_{max}$.

In recent years, a body of research has addressed the scheduling problem on PBPMs. [1] gave a mathematical model and three heuristics to minimize the makespan under arbitrary job ready times. [2] proposed a greedy randomized adoptive search procedure approach, developed a lower bound on the makespan, and compared the results against those reported by the heuristic proposed by [1]. [3] provided a mixed-integer programming model for PBPMs with non-identical job sizes and dynamic job arrivals. They still proposed two meta-heuristic algorithms based on SA and GA, which were combined with a dynamic programming algorithm. [4] investigated the same PBPMs problem considered in this research. They combined a GA algorithm with the best-fit heuristic to form batches first and employed a dispatching rule to schedule the formed batches second. [5] was the first to provide several heuristics and a branch-and-bound algorithm for solving BPM with non-identical sizes. To schedule a batch-processing machine for minimizing C_{max} with different job sizes, [6] and [7] proposed two different algorithms: a simulated annealing approach and a genetic algorithm, respectively. [8] proposed two different genetic algorithms based on different encoding schemes to minimize makespan with non-identical job sizes.

Given the special case on a single BPM ($1|r_i, s_i, p\text{-}batch|C_{max}$) is strongly $\mathcal{NP}$-hard [9], hence the target problem in this paper is $\mathcal{NP}$-hard as well. Generally, solving PBPMs typically consists of two interrelated decisions involving batch formation and batch scheduling, which makes the development of efficient algorithms for this problem tough and challenging. Different from classical algorithms using the "batch first - schedule second" policy to construct a solution, we introduced a new construction policy which will enable the two decisions to utilize information with each other during the solution construction process. Based on this, we developed an ant colony optimization algorithm (ACO) to solve the target problem.

2 Proposed Ant Colony Optimization Algorithm

In recent years, because of the stochastic and iterative nature of meta-heuristic, more and more meta-heuristic algorithms are applied for different combinatorial optimization problems. The ACO framework is a novel nature inspired meta-heuristic introduced by Macro Dorigo in the early 1990s and has gained immense popularity in recent years.

2.1 Pheromone Trails and Heuristic Information

In the case of our problem, since we construct a feasible solution by assigning a job into the current batch iteratively, we require the value of $\tau_{b,u}$ to refer to the

desirability of assigning the next job u into the current batch b on machine m. In this paper, pheromone value $\tau_{u,j}$ represents the current pheromone trail defined as the desirability of having both job u and job j in the same batch, where job j is the one that has been assigned into the current batch b on machine m as the Eq. (1) below.

$$\tau_{bm}^{u} = \begin{cases} \frac{\sum_{j \in bm} \tau_{u,j}}{|bm|} & \text{if } bm \neq \emptyset \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

The concept of Waste and Idle Space ($\mathcal{WIS}$) was first proposed by [9] for scheduling a single BPM. In this paper, before given the definition of heuristic information, the extended definition of $\mathcal{WIS}$ is presented first for the new context of PBPMs scheduling as follows.

Definition 1. *Suppose π be a feasible schedule and π_m be a sub-schedule of π on machine m. Let b be an arbitrary batch on machine m of the sub-schedule π_m. $\mathcal{WIS}(\pi_m)$ is the Waste and Idle Space of sub-schedule π_m, which equals to the sum of the Waste Space ($\mathcal{WS}(\pi_m)$) and the Idle Space ($\mathcal{IS}(\pi_m)$) of sub-schedule π_m. $\mathcal{WIS}(\pi)$ is the Waste and Idle Space of the schedule π, which equals to the maximum value of all sub-schedules in the schedule π.*

The PBPMs scheduling problem is a multi-dimensional problem, which involves the constraints of non-identical job sizes, unequal release times, arbitrary processing times. Inherently, The concept of $\mathcal{WIS}$ is developed by considering the multi-dimensions simultaneously, which can be calculated by the following expression:

$$\mathcal{WIS}(\pi_m) = B \cdot C_m - \sum_{b \in m} \sum_{j \in bm} s_j \cdot p_j \tag{2}$$

Based on the definition of $\mathcal{WIS}$, [9] pointed out that the smaller $\mathcal{WIS}(\pi)$, the better the solution for $1|r_i, s_i, p\text{-}batch|C_{max}$. In our case, although the processing environment involves several parallel batch-processing machines instead of a single one, the optimal objective of makespan for both cases is the same. Minimizing makespan on PBPMs is equal to minimizing maximum makespan for each BPM. Therefore, we can try to batch jobs together to reduce the $\mathcal{WIS}(\pi_m)$ as much as possible during the process of batch formation. Hence, the heuristic information of the ACS algorithm can be calculated as follows:

$$\eta_{bm}^{u} = \begin{cases} s_u \cdot p_u - B \cdot (C_{m'} - C_m) & \text{if } s_u \cdot p_u - B \cdot (C_{m'} - C_m > 0 \\ \frac{1}{s_u \cdot p_u - B \cdot (C_{m'} - C_m) + 1} & \text{otherwise} \end{cases} \tag{3}$$

where $C_{m'}$ denotes the new value of C_m after job u has been assigned into the sub-schedule π_m. It means the more reduced value after job u has been assigned, the more desirability of having job u in batch b on machine m.

Note that, due to the different construction feature in our approach, we can obtain the machine completion time during the process of batch formation so

that the $\mathcal{WIS}(\pi_m)$ of a machine can be employed, which makes the calculation of $\mathcal{WIS}$ more reasonable.

2.2 Candidate List

Considering the constraints of capacity and release time in our case, the candidate list strategy is necessary especially for large scale problems. Intuitively, the capacity constraint is the only mandatory constraint. That is to say, any unscheduled job under the capacity constraint has the possibility to be selected as a candidate job.

However, such a candidate strategy does not take the constraint of release time into account. As we know, job release time is a key factor need to be considered during the process of solution construction. If a candidate job has arrived by the batch starting time, then the release time of the job can be ignored. Otherwise, it may delay the process of the current batch and result in an idle time for the current machine. Thus, it is necessary to find out the candidate condition that can provide a better objective value even if the process of the current batch is delayed.

Theorem 1. *Let b be the last batch on machine m with at least one job, u be an unscheduled job whose release time is larger than the starting time of batch b. If job u satisfies the capacity constraint and candidate condition (4), then the solution assigning job u into batch b is better than the one processing job u after batch b.*

$$r_u \leq S_{bm} + min\{p_u, P_{bm}\} \tag{4}$$

Proof. By contradiction. There are two cases depending on whether job u is assigned into batch b or not.
(a) In the case of assigning job u into batch b, then the objective value will be $C_{max1} = r_u + max\{p_u, P_{bm}\}$.
(b) In the case of processing job u after batch b, then the objective value will be $C_{max2} = S_{bm} + p_u + P_{bm}$.

According to the relationship between p_u and P_{bm}, there are two situations need to be considered. If $p_u > P_{bm}$, then $C_{max1} = r_u + p_u$. Suppose $C_{max1} > C_{max2}$, thus it demands $r_u - S_{bm} > P_{bm}$, which is contradictory to Eq. (4). Otherwise, if $p_u < P_{bm}$, then $C_{max1} = r_u + P_{bm}$. Suppose $C_{max1} > C_{max2}$, thus it demands $r_u - S_{bm} > p_u$, which is contradictory to Eq. (4) either.

Theorem 1 presents the candidate condition for yielding a better objective value if the batch process is delayed. Based on the Theorem 1, we propose the candidate strategy CL used in the ACS algorithm as follows.

$$CL = \left\{u | s_u \leq (B - \sum_{j \in bm} s_j) \text{ and } r_u \leq S_{bm} + min\{p_u, P_{bm}\}\right\} \tag{5}$$

The strategy CL can make a compromise between increasing utilization of machines and decreasing delay time of waiting jobs, which is able to achieve a tradeoff between exploitation and exploration in the ACS algorithm.

2.3 Solution Construction Policy

As mentioned before, the classical policy used in a meta-heuristic always constructs a final solution in a hierarchical manner (*i.e.*, batch formation decision precedes the batch scheduling decision). Due to the new construction policy, the proposed ACS algorithm builds a feasible solution by considering the two interrelated decisions simultaneously and interacting. At first, each ant of the colony selects the first available machine randomly as the current machine, and an empty batch is established on it. If there are arrived jobs before the current machine is available, assign a random job into the empty batch from the arrived ones. Otherwise, put the earliest job from the unscheduled jobs into the batch. Once the current batch contained with at least one job, each ant chooses a next job from a candidate list at a time to continue the batch formation process iteratively by using a pesudo-random-proportional rule. As soon as the current batch is full or if there are no more jobs which can fit the candidate conditions of the batch being formed, then the batch is closed and scheduled on the current machine. Repeat the formation and scheduling process until all jobs are scheduled.

The pesudo-random-proportional rule that can be stated as follows:

$$u = \begin{cases} \arg\max\limits_{u \in CL} [\tau_{bm}^{u}] \cdot [\eta_{bm}^{u}]^{\beta} & \text{if } q \leq q_0 \\ \hat{u} & \text{otherwise} \end{cases} \tag{6}$$

where q is a random number uniformly distributed in [0,1], q_0 is a parameter to be set by the user representing the probability that the job is chosen which gives the highest aggregate value of pheromone and heuristic. The random variable $\hat{u}$ is selected according to the probability distribution given:

$$\hat{u} = \begin{cases} \frac{[\tau_{bm}^{u}] \cdot [\eta_{bm}^{u}]^{\beta}}{\sum_{u \in CL} \left([\tau_{bm}^{u}] \cdot [\eta_{bm}^{u}]^{\beta} \right)} & \text{if } u \in CL \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Parameter β determines the relative influence of the heuristic information for the objective of C_{max}. The probability of choosing a job outside CL is zero.

2.4 Update of Pheromone Trails

Global Updating Rule: This rule is executed only to the best-so-far solution provided after each iteration.

$$\tau_{u,j} = (1 - \rho) \cdot \tau_{u,j} + \rho \cdot \Delta\tau_{u,j} \tag{8}$$

where $\rho \in (0,1)$ is a parameter to control the speed of pheromone evaporation. Pheromone is deposited by a quantity $\Delta\tau_{u,j} = 1/C_{max}$.
Local Updating Rule: Different from the global updating rule, this rule is performed to pheromone matrix immediately after each ant has added a job to a batch during the construction process.

$$\tau_{u,j} = (1-\rho) \cdot \tau_{u,j} + \rho \cdot \tau_0 \tag{9}$$

where τ_0 is set to the same as the initial value of pheromone trails. The local updating rule allows an increase in the exploration of arcs that have not been visited yet.

3 Computational Experiments

3.1 Experimental Design

Random instances were generated in a way similar as in [4]. First, we generated random problem instances for $n = 20, 50, 100$, and 200 jobs, respectively. After specifying the number of jobs in a given instance, for each job j, an integer processing time p_j was generated from the uniform distribution [1,10] and an integer size s_j was generated from the uniform distribution [1,10], [2,4], and [4,8]. In addition, the following equation was used to determine the maximum release time.

$$r_{max} = R\frac{E(p)E(s)|J|}{B|M|} \tag{10}$$

where $E(p)$ and $E(s)$ are the mathematical expectation of processing time and size, and the factor R determines the relative frequency of jobs arrivals. In this paper, release time r_j was generated from uniform distribution [0,r_{max}] and $R = 0.5, 1.0$. The number of machines and maximum capacity of a machine was fixed to 2 and 10 for all instances. There are a total of 24 categories of problems with different combinations. Five instances were generated for each category. Thus, totally 120 problem instances were used in our experiments.

Table 1. Parameter settings

λ	q_0	β	ρ	I_{max}	τ_0
15	0.9	4	0.1	100	$1/LB$

The performance of the ACS algorithm is generally sensitive to the parameter setting which influences the search performance and the convergence quality. We incorporate the parameters used in the literature [10] and conducted several pilot experiments to determine the relatively important parameters. The final parameters values selected for the ACS algorithm were presented in Table 1.

3.2 Computational Results and Comparisons

A computational study was conducted to evaluate the efficiency and effectiveness of the proposed algorithm, which was coded in C++ and executed on a PC with a processor (2.20 GHz CPU speed) and 2 GB RAM.

We employed a heuristic BE and a meta-heuristic GA proposed by [4] as comparison algorithms. Table 2 presents the average results obtained over five randomly generated instances for each problem configuration. In the table, for two algorithms GA and ACO, each instance was tested ten times and the mean performance and runtime were reported. Δ_1 and Δ_2 were calculated by $\Delta_1 = GA - ACS$ and $\Delta_2 = BE - ACS$ to provide the difference between the algorithms.

Table 2. Comparison of BE, GA, and ACS for different categories

Run Code	BE	GA	Time	ACS	Time	Δ_1	Δ_2	Run Code	BE	GA	Time	ACS	Time	Δ_1	Δ_2
J20S1R1	41.4	40.0	4.6	40.0	2.7	0.02	1.45	J100S1R1	172.2	170.3	30.5	168.5	98.7	1.82	3.67
J20S1R2	46.9	44.0	3.9	45.1	2.6	-1.01	1.86	J100S1R2	202.2	199.0	32.8	185.7	112.3	13.24	16.43
J20S2R1	27.1	26.5	2.7	25.5	3.0	0.93	1.54	J100S2R1	107.7	104.7	16.0	95.8	100.0	8.89	11.87
J20S2R2	34.5	32.0	2.8	28.8	2.9	3.23	5.64	J100S2R2	125.9	122.1	15.4	103.2	98.0	18.86	22.63
J20S3R1	45.1	44.6	4.0	45.8	2.5	-1.18	-0.77	J100S3R1	199.2	198.6	35.7	197.7	114.7	0.90	1.53
J20S3R2	50.0	49.2	3.8	49.0	2.5	0.21	1.00	J100S3R2	218.9	217.0	34.2	214.3	118.2	2.74	4.58
Average	41.5	40.0	3.6	39.3	2.7	0.68	2.20	Average	173.8	170.9	27.4	160.9	107.0	10.06	12.94
J50S1R1	91.0	88.9	9.1	89.0	16.9	-0.07	2.02	J200S1R1	328.6	327.0	84.7	320.6	688.6	6.36	7.96
J50S1R2	109.0	105.7	9.4	100.4	19.3	5.30	8.56	J200S1R2	398.5	396.3	72.8	358.4	684.9	37.89	40.09
J50S2R1	54.1	52.8	6.0	48.7	17.8	4.18	5.42	J200S2R1	207.5	203.1	57.8	180.6	626.8	22.46	26.87
J50S2R2	66.7	61.3	6.1	54.9	17.3	6.41	11.80	J200S2R2	259.4	247.1	53.2	200.8	557.8	46.32	58.64
J50S3R1	99.3	98.0	11.1	96.7	18.5	1.32	2.64	J200S3R1	406.7	406.7	91.5	408.5	759.8	-1.79	-1.79
J50S3R2	120.8	118.4	10.1	118.3	19.5	0.08	2.42	J200S3R2	431.6	428.8	95.0	420.2	796.4	8.69	11.43
Average	91.6	88.6	8.6	84.8	18.2	3.83	6.83	Average	346.9	341.5	75.8	315.2	685.7	26.25	31.68

As can be seen from Table 2, ACS outperforms the other three algorithms while BE has the worst overall performance for all categories of test problems. As the number of jobs increases, the Δ_1 and Δ_2 are improved from 0.68 and 2.20 for category J20 up to 26.25 and 31.68 for category J200, respectively, which demonstrates that ACS provide superior results compared to BE and GA with the increasing problem scale. Because as problems are smaller, the solution space is also limited. In this condition, ACS performs slightly better than GA and BE but the difference is not very meaningful. However, when dealing with large scale problems, there are more decisions to be made and consequently large solution spaces to be explored. In this case, due to the exponential growth of the solution space, the comparison algorithms become less efficient. Because of the adoption of the new construction policy and the introduction of candidate strategy, the ACS algorithm has the ability to explore potentially better solutions in large search spaces. Consequently, ACS can provide an evident superiority compared to the GA and BE algorithms for finding solutions in large solution spaces.

Considering the computational efficiency of the two meta-heuristics, it is observed that ACS takes less time than GA to find a better solution for a majority of the instances with 20 jobs. The computational time of ACS increases rapidly with the number of jobs when compared to GA. As can be seen from Table 2,

the computational time of ACS is higher than GA for all of the instances as the number of jobs is great than 50. Due to distinctive encoding mechanisms and search techniques, ACS has the capability to reduce the search space significantly and to obtain better solutions. Consequently, even if the computational cost is higher than GA, it is still kept at a reasonable level considering the solution quality.

4 Conclusion

In this paper, we addressed the PBPMs problem considering non-identical job sizes and dynamic job arrivals for minimizing makespan. Based on the introduced new construction policy and candidate list strategy, an ACS algorithm was proposed to solve this PBPMs problem. Results showed that the proposed ACS algorithm was capable of obtaining better solutions compared to the existing algorithms, especially for solving problems with large search spaces.

References

1. Chung, S., Tai, Y., Pearn, W.: Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. International Journal of Production Research 47(18), 5109–5128 (2009)
2. Damodaran, P., Vlez-Gallego, M., Maya, J.: A grasp approach for makespan minimization on parallel batch processing machines. Journal of Intelligent Manufacturing 22, 767–777 (2011)
3. Wang, H.M., Chou, F.D.: Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics. Expert Systems with Applications 37(2), 1510–1521 (2010)
4. Chen, H., Du, B., Huang, G.Q.: Metaheuristics to minimise makespan on parallel batch processing machines with dynamic job arrivals. International Journal of Computer Integrated Manufacturing 23(10), 942–956 (2010)
5. Uzsoy, R.: Scheduling a single batch processing machine with nonidentical job sizes. International Journal of Production Research 32(7), 1615–1635 (1994)
6. Melouk, S., Damodaran, P., Chang, P.Y.: Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. International Journal of Production Economics 87(2), 141–147 (2004)
7. Damodaran, P., Manjeshwar, P.K., Srihari, K.: Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms. International Journal of Production Economics 103(2), 882–891 (2006)
8. Kashan, A., Karimi, B., Jolai, F.: Effective hybrid genetic algorithm for minimizing makespan on a single-batch-processing machine with non-identical job sizes. International Journal of Production Research 44(12), 2337–2360 (2006)
9. Xu, R., Chen, H., Li, X.: Makespan minimization on sigle batch-processing machine via ant colony optimization. Computers & Operations Research 39(6), 582–593 (2012)
10. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)

Understanding Instance Complexity in the Linear Ordering Problem

Josu Ceberio, Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group,
Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU
Donostia, 20018 Gipuzkoa, Spain
{josu.ceberio,leticia.hernando,alexander.mendiburu,ja.lozano}@ehu.es

Abstract. The Linear Ordering Problem is a combinatorial optimization problem which has been frequently addressed in the literature due to its numerous applications in diverse fields. In spite of its popularity, little is known about its complexity. In this paper we analyze the linear ordering problem trying to identify features or characteristics of the instances that can provide useful insights into the difficulty of solving them. Particularly, we introduce two different metrics, *insert ratio* and *ubiquity ratio*, that measure the difficulty of solving the LOP with local search type algorithms with the insert neighborhood system. Conducted experiments demonstrate that the proposed metrics clearly correlate with the complexity of solving the LOP with a multistart local search algorithm.

Keywords: linear ordering problem, instance, complexity, local optima.

1 Introduction

The Linear Ordering Problem (LOP) is a classical combinatorial optimization problem which has received the attention of the research community since it was studied for the first time by Chenery and Watanabe [1]. Due to its numerous applications in diverse fields such as archeology [4], economics [7] or mathematical psychology [6], a wide variety of optimization strategies have been proposed in the literature. As proof of this, Marti and Reinelt [8] presented a review of the most successful exact and heuristic algorithms, including branch and bound, constructive heuristics, local searches or variable neighborhood search. Garey and Johnson [3] demonstrated that the LOP is a *NP-hard* problem, which means that there is no known algorithm able to solve up to the optimality all LOP instances in polynomial time. However, as seen for most of the combinatorial optimization problems, the difficulty of solving the instances varies with the size of the instance, and other additional unknown parameters. In the particular case of the LOP, one can easily find instances of large size that are easier to solve than other of smaller size for a wide set of algorithms.

In this regard, the community has also tried to extract characteristics from the instances that could be used to measure their difficulty or to provide additional

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 479–486, 2013.

information to the algorithm used for solving it. For instance, Schiavinotto and Stutzle [9] analyzed LOP instances from different benchmarks, extracting features that later were used to rank the benchmarks according to their difficulty, and proposed new algorithms for optimizing the LOP. However, we think that the properties they studied could be further developed.

In this work, we investigate this direction trying to identify features or characteristics of the LOP instances that influence the complexity of solving LOP instances with local search type algorithms. Particularly, we introduce two different metrics: *insert ratio* and *ubiquity ratio* that give a measure of the difficulty of achieving the optimal solution with an insert neighborhood system.

The rest of the paper is organized as follows: in the following section, the definition of the LOP is given. In Section 3, the instance complexity of the LOP is studied by introducing the insert ratio and ubiquity ratio. Next, in Section 4, some experiments are run in order to evaluate the validity of the proposed metrics. Finally, some conclusions and ideas for future work are presented in Section 5.

2 The Linear Ordering Problem

Given a matrix $B = [b_{ij}]_{n \times n}$ of weights, the LOP consists of finding a simultaneous permutation σ of the rows and columns of B, such that the sum of the weights above the main diagonal is maximized. The equation below formalizes the LOP function:

$$f(\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{\sigma_i \sigma_j}$$

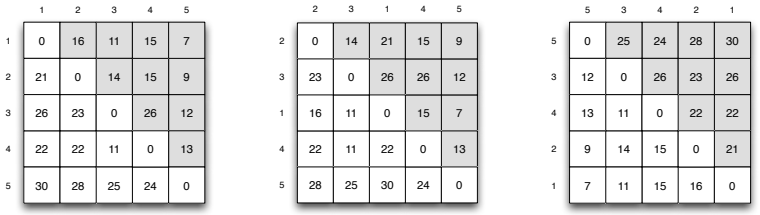
where σ_i denotes the index of the row (and column) ranked in position i in σ.

Let us consider an example of 5×5 LOP instance (see Fig. 1). The initial matrix is represented by the identity permutation $\sigma = (1, 2, 3, 4, 5)$ (see Fig. 1a), and the associated fitness $f(\sigma)$ is 138. The solution $\sigma' = (2, 3, 1, 4, 5)$ introduces a different ordering of the indices that provide a better solution than σ (see Fig. 1b). The optimal solution for this example is given by $\sigma^* = (5, 3, 4, 2, 1)$ (see Fig. 1c).

For the sake of studying the LOP function, we analyze separately the contribution of each index of the solution to the objective function. When an index j is ranked in position i, $\sigma_i = j$, the contribution of the index j to the objective function is given by the sum of weights of the column j in the rows $\sigma_1, \ldots, \sigma_{i-1}$ and weights of the row j in the columns $\sigma_{i+1}, \ldots, \sigma_n$. Formally it is expressed as

$$f_\sigma(\sigma_i) = \sum_{z=1}^{i-1} b_{\sigma_z \sigma_i} + \sum_{z=i+1}^{n} b_{\sigma_i \sigma_z}$$

Note that the rank i indicates only the number of weights of the corresponding column $\{1, \ldots, i-1\}$ and row $\{i+1, \ldots, n\}$ that will be considered in the sum. The specific row and column weights come given by the previously ranked $i-1$



	1	2	3	4	5
1	0	16	11	15	7
2	21	0	14	15	9
3	26	23	0	26	12
4	22	22	11	0	13
5	30	28	25	24	0

(a) $\sigma = (1, 2, 3, 4, 5)$ $f(\sigma) = 138$.

	2	3	1	4	5
2	0	14	21	15	9
3	23	0	26	26	12
1	16	11	0	15	7
4	22	11	22	0	13
5	28	25	30	24	0

(b) $\sigma' = (2, 3, 1, 4, 5)$ $f(\sigma') = 158$.

	5	3	4	2	1
5	0	25	24	28	30
3	12	0	26	23	26
4	13	11	0	22	22
2	9	14	15	0	21
1	7	11	15	16	0

(c) $\sigma^* = (5, 3, 4, 2, 1)$ $f(\sigma^*) = 247$.

Fig. 1. Example of a 5×5 LOP instance. The configuration of the matrix of weights for three different solutions is described.

indices $\{\sigma_i, \ldots, \sigma_{i-1}\}$ and the posterior $n-i+1$ indices $\{\sigma_{i+1}, \ldots, \sigma_n\}$. However, note that the contribution of an index is independent to the ordering of the previously ranked indices and to the posterior indices.

Following the previous example, the contribution of the index 4 in σ (see Fig. 1a) consist of the sum of the weights associated to this index in the upper triangle (15+15+26+13). If we check the contribution of the index 4 in σ' (see Fig. 1b), we see that the contribution of the index remains equal regardless of what the ordering of the indices is in the previous $\{1, 2, 3\}$ positions and in the posterior positions $\{5\}$.

3 Analysis of Complexity

As previously mentioned, in this work we aim to study the properties of the instances that affect the complexity of local search type algorithms. Since many works in the literature [9,2] clearly state that the insert neighborhood is the most appropriate system for solving the LOP with a local search algorithm, our study will be carried out for this neighborhood.

In the following lines, the *insert ratio* and *ubiquity ratio* are introduced in detail.

3.1 Insert Ratio

The insert ratio measures the intercalation between the weights of the column and of the row associated to an index. It is calculated as follows:

1. Put all row and column weights in a vector and order these numbers.
2. For each pair of consecutive weights belonging to the column, calculate the distance
3. Repeat the same process as before for the weights of the row.
4. The insert ratio is calculated by summing the obtained distances, and dividing by $n - 1$.

The overall insert ratio of the instance is calculated by averaging the insert ratio of each of the indices. Fig. 2 illustrates the insert ratio calculated for the index 4 of the matrix.

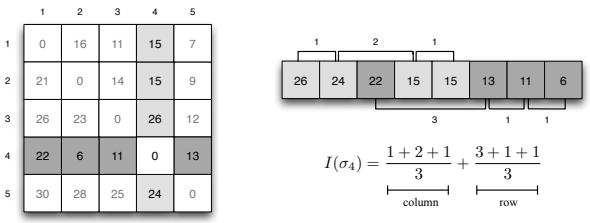


Fig. 2. Insert ratio of the index σ_4

Given a specific index i, when all the weights of the column i are higher than the weights of the row i, the index i can only generate local optima in the last position of σ, and only in the first position when the opposite scenario occurs. The indices that agree with the exposed cases receive very low insert ratios, suggesting that this type of indices are easy to rank. When the overall insert ratio associated to an instance is low, the instance is supposed to be easy to solve. On the contrary, a high insert ratio, suggests that the indices are more susceptible to generate local optima in many ranks of σ, and thus, the difficulty of the instance is higher.

3.2 Ubiquity Ratio

The ubiquity ratio measures the percentage of locations where the indices are susceptible to generate local optima using the insert neighborhood. An index may generate local optima in a certain position if all the possible insert operations over the index decreases its contribution. Note that for each index i, the weights b_{ij} and b_{ji} for $j = \{1, \ldots, n\}$ appear always in symmetric positions with respect to the main diagonal, i.e. they never appear together in the upper triangle. Therefore, an index i may generate a local optima when ranked in position j if there exists an ordering σ such that the following two constraints are true.

$$\sum_{z=1}^{j-1} b_{\sigma_z,\sigma_j} - b_{\sigma_j,\sigma_z} > 0 \wedge \sum_{z=j+1}^{n} b_{\sigma_z,\sigma_j} - b_{\sigma_j,\sigma_z} < 0$$

In order to determine the ranks where the indices may generate local optima, we count the positions where the index ranked in position j cannot generate a local optima, that is to check whether the sum of all the differences in positions $\{1, \ldots, j-1\}$ is negative and the sum of all the differences in positions $\{j+1, \ldots, n\}$ is positive. The ubiquity ratio of an index is calculated by dividing the number of positions where the index may generate local optima, with the total

number of positions. The global ubiquity ratio is then calculated by averaging the ratios of each of the indices. Fig. 3 illustrates the procedure of calculating the ubiquity ratio of the index 4.

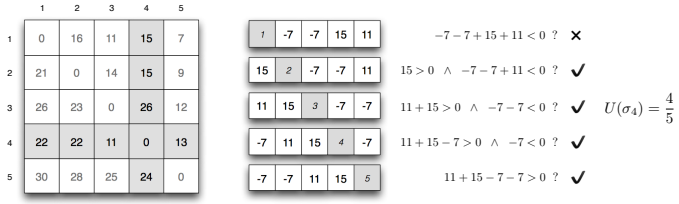


Fig. 3. Ubiquity ratio of the index σ_4

4 Experimentation

In order to confirm the validity of the metrics proposed, we introduce two different set of experiments. Additionally, two artificially generated benchmarks of instances are used.

Benchmarks A and B of instances. 540 instances of sizes $\{8, 9, 10, 11, 12, 13\}$ (benchmark A) and 600 instances of sizes $\{20, 30, 40, 50, 60, 70\}$ (benchmark B) have been generated following the next procedure. We start by generating the weights associated to the first index, then the second and so on. Note that when we generate the weights for the first index, we have also indirectly generated some of the weights for the rest of the indices, and the same for the second index and so on. In order to generate the weights for index i, we uniformly at random sample a vector of $2(n-i-1)$ weights in the range [0,999]. Next, we order these weights. After that, following the order of the weights, we make groups of size t (t is a parameter of the procedure that ranges from 0 to 10). Then, we randomly decide where, column or row, to place each pair of groups (one to the row and the other to the column).

Experiment 1. We analyze the correlation of the insert and ubiquity ratios with respect to the estimated number of local optima for the benchmark B of LOP instances. Particularly, the benchmark B of instances is used. Following the recommendations of a recently published review on local optima estimation methods [5], we have chosen two methods to estimate the number of local optima of the instances: *ChaoBunge* and *ChaoLee2*.

Fig. 4 and 5 introduce the results for the proposed metrics according to the average of 10 repetitions of *ChaoLee2* and *ChaoBunge*. Results show a good correlation between the *ubiquity* and *insert* ratios and the number of local optima. In fact, the higher the value of the metric, the higher the number of local optima.

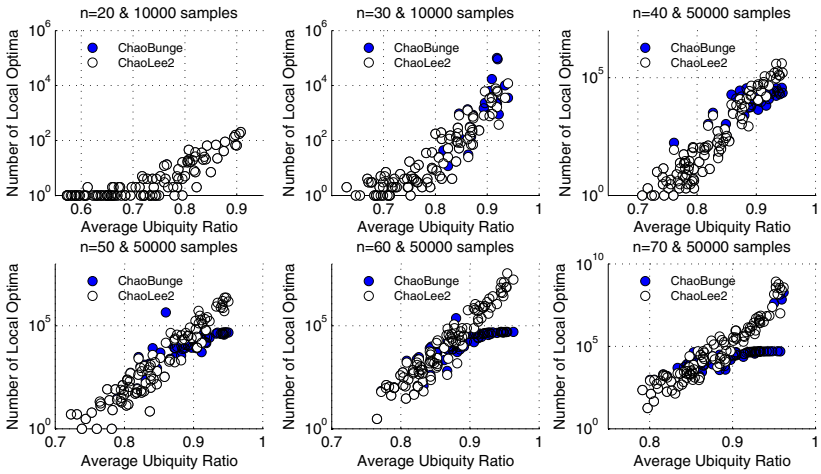


Fig. 4. The estimated number of local optima related to the average *ubiquity* ratio of the instance.

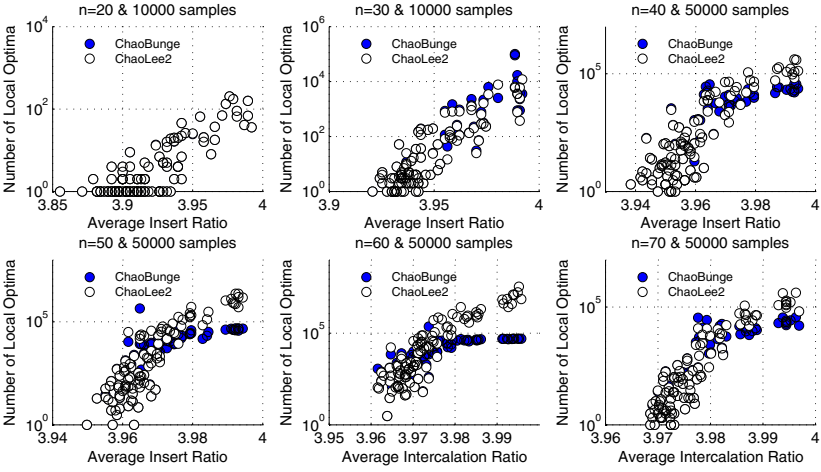


Fig. 5. The estimated number of local optima related to the average *insert* ratio of the instance

Experiment 2. In this experiment, we run a muti-start local search greedy algorithm (MLS) with the insert neighborhood, and we compute the number of evaluations needed to achieve the optimal solution of the benchmark A instances.

Fig. 6 and 7 introduce the results for the proposed metrics according to the average of 10 repetitions of the MLS. Results show that the higher the *ubiquity* and *insert* ratios, the higher the number of evaluations performed. This behavior is especially evident for instances of size $\{11, 12, 13\}$.

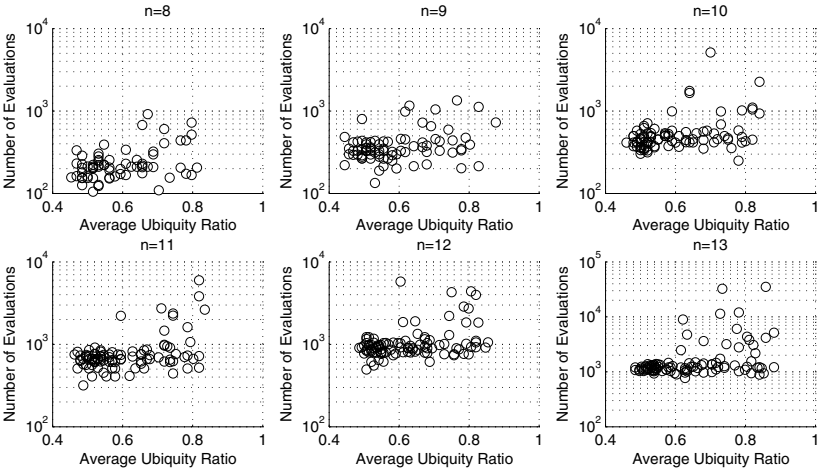


Fig. 6. The average number of evaluations performed by the MLS related to the average *ubiquity* ratio of the instance.

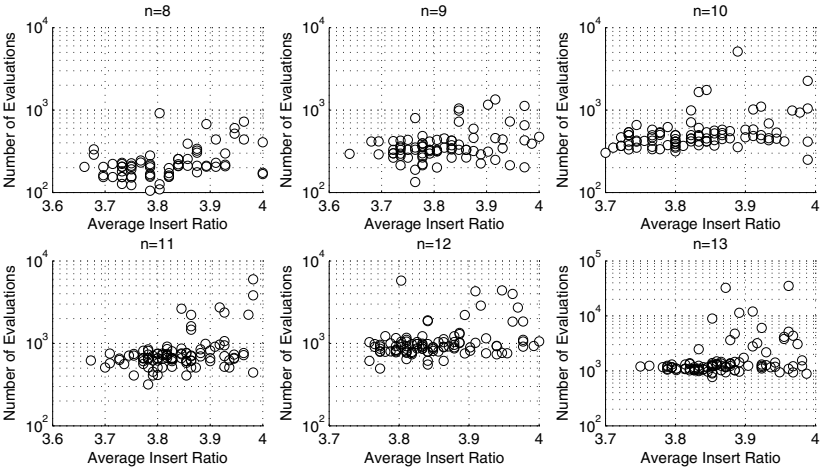


Fig. 7. The average number of evaluations performed by the MLS related to the average *insert* ratio of the instance

5 Conclusions and Future Work

In this paper, a preliminary analysis of the instance complexity of the LOP was presented. Particularly, we focused on identifying the characteristics of the instances that influence the complexity of solving them with local search type algorithms with the insert neighborhood. Characterizing the conditions required by each index in the solution to generate local optima for the insert neighborhood, we presented two new metrics, *insert ratio* and *ubiquity ratio*. Conducted

experiments showed a correlation between the proposed metrics and the complexity of solving an artificially generated set of LOP instances with multistart local search algorithms.

As future work, it would be interesting to extend the proposed metrics with pairwise index relations or even larger ones. Alternatively, the influence of ties of pairs of weights to the number of local optima should be studied, which, at first sight, introduce a high redundancy in the search space.

Acknowledgments. This work has been partially supported by the Saiotek and Research Groups 2013-2018 (IT-609-13) programs (Basque Government), TIN2010-14931 (Ministry of Science and Technology), COMBIOMED network in computational bio-medicine (Carlos III Health Institute), and by the NICaiA Project PIRSES-GA-2009-247619 (European Commission). Josu Ceberio and Leticia Hernando hold a grant from Basque Government.

References

1. Chenery, H.B., Watanabe, T.: International comparisons of the structure of production. Econometrica 26(4), 487–521 (1958)
2. Garcia, C.G., Pérez-Brito, D., Campos, V., Martí, R.: Variable neighborhood search for the linear ordering problem. Comput. Oper. Res. 33(12) (2006)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
4. Glover, F., Klastorin, T., Klingman, D.: Optimal weighted ancestry relationships. Management science report series. University of Colorado (1972)
5. Hernando, L., Mendiburu, A., Lozano, J.A.: An evaluation of methods for estimating the number of local optima in combinatorial optimization problems. In: Evolutionary Computation (2013)
6. Kemeny, J.G.: Mathematics without numbers. Daedalus 88, 577–591 (1959)
7. Leontief, W.: Input-Output Economics. Cambridge University Press (2008)
8. Martí, R., Reinelt, G.: The linear ordering problem: exact and heuristic methods in combinatorial optimization, vol. 175. Springer (2011)
9. Schiavinotto, T., Stützle, T.: The linear ordering problem: instances, search space analysis and algorithms. Journal of Math. Modelling and Algorithms (2004)

Multi-Objective Evolutionary Algorithm Based on Decomposition for Air Traffic Flow Network Rerouting Problem

Xiao Zhang, Mingming Xiao, and Miao Zhang

School of Electronic and Information Engineering, Beihang University, Beijing, China
343761380@qq.com

Abstract. Air Traffic Flow Network Rerouting Problem (ATFNRP), which aims to alleviate the flight delays caused by the increasing traffic and extreme weather, has become more and more serious in air traffic flow management. This paper proposes a multi-objective general rerouting model considering both total delay cost and airlines fairness and adopts Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) for ATNFRP. Empirical studies using the real data of China airspace demonstrate that MOEA/D outperforms or performs similarly to three well-acknowledged Multi-Objective Evolutionary Algorithms (MOEAs) on ATFNRP.

Keywords: Air Traffic Flow Network Rerouting Problem (ATFNRP), Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), penalty function.

1 Introduction

With rapid development of civil aviation, the growing traffic flow and limited airspace capacity have resulted in severe flight delays. How to reduce the delays effectively and fairly has become critical in the field of Air Traffic Flow Management (ATFM). Air Traffic Flow Network Rerouting Problem (ATFNRP) which is to find optimal departure time-slots and rerouting paths for all flights involved in the air traffic flow network is a general rerouting problem arising in the field of ATFM.

During recent years, many scholars have been carrying out research on ATFNRP. Bertsimas and Patterson [1-2] first analyzed rerouting problem from a general perspective. They established a dynamic multi-commodity network flow model to minimize the delay cost with the constraints of sectors capacity and adopted an integrated mathematical programming approach to solve it. Inspired by the work, many approaches have been carried out to solve the above model, including heuristic searching [3] considering flights' priorities, genetic algorithms [4] and multi-objective micro genetic algorithm [5].To formulate ATFNRP exactly, Bertsimas, Lulli and Odoni [6] presented an integer programming model which covers all the phases of each flight and solves for an optimal combination of flow management actions, including ground-holding, rerouting, speed control, airborne-holding, subsequent flights and

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 487–496, 2013.

cancellation. Besides, Wan and Tian [7] took the interests of airlines into account and established a multi-objective model with evolutionary algorithm employed.

Currently, most works regard the flight delays as the sole objective while taking airlines fairness as constraints and seeking solutions with integer optimization which can neither achieve desired fairness nor satisfy the needs of efficiency. However, ATFNRP is a multi-objective problem which is high-dimensional, non-separable and multi-constrained essentially. And the optimization process is supposed to seek different trade-off between total delay cost and airlines fairness efficiently.

Therefore, the goal of this paper is to make deep research on ATFNRP in the context of real-world ATFM. A multi-objective general rerouting model is established taking both total delay cost and airlines fairness as the bi-objectives. The model considers the constraints of segments and airports capacity to be more realistic and exact. Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) with dynamic penalty function and normalization is applied to reduce the computational complexity and accelerate the evolution according to the problem characteristics.

The remainder of this paper is organized as follows: Section 2 presents the mathematical description of ATFNRP. Section 3describes the MOEA/D for solving ATFNRP in detail. Experimental study is presented in Section 4 to evaluate the efficacy and efficiency of MOEA/D for ATFNRP and analyze dynamic penalty function. Section 5 concludes the paper.

2 Mathematical Model

In this section, we will present a mathematical model of ATFNRP in detail.

The notations and its definitions of main input data are shown as follows. F, A, L, Seg and $Airport$ are the sets of flights, airlines, aircraft types, segments and airports respectively. Sets of feasible routes and departure time-slots for flight f are denoted as R_f *and* T_f. A pair of decision variable (r_f, τ_f) is associated with each flight f, where r_f represents the selected route and τ_f is the departure time-slot.

2.1 Objectives

The two conflicting objectives for ATFNRP model are minimization of Total Delay Cost and maximization of Airlines Fairness [8]. Next we will describe the two objectives in detail respectively.

Objective 1: minimization of the total delay cost (TDC)

The TDC of flights represents the economic cost, and it is calculated as equation (1).

$$\begin{aligned} \min TDC: f_1 &= \min \sum_{f \in F} (C_f^{GH} + C_f^{ED} + C_f^{S}) \\ &= \min \sum_{f \in F} (\lambda_{\alpha,l}^{GH} \cdot (\tau_f - \tau_{f0}) + \lambda_{\alpha,l}^{ED} \cdot \left(\tau_f - \tau_{f0}\right) + \lambda_{\alpha,l}^{S} \cdot y_f (\tau_f^{ar} - \tau_{f0}^{ar})) \end{aligned} \tag{1}$$

Where C_f^{GH} and C_f^{ED} denote the ground-holding and rerouting delay cost separately and C_f^S is the impact cost of the subsequent flight f'. And, $\lambda_{\alpha,l}^{GH}$, $\lambda_{\alpha,l}^{ED}$ and $\lambda_{\alpha,l}^{S}$ are their delay coefficients for flight f of airline α, aircraft type l. When $y=0$, subsequent flight is not existed. And, the sum of them is the total delay cost of flight f without consideration of cancelation and air holding.

Objective 2: maximization of the airlines fairness (AF)

The airlines fairness is represented by the difference between the growth rate and the expectation of operation cost and can be calculated as equation (2).

$$\max AF: f_2 = \min(\sum_{\alpha \in A} \lambda_\alpha \mid E_\alpha - \overline{E} \mid) \tag{2}$$

Where λ_α is the weight of Airline α and is reflected by the ratio of number of flights for Airline α to total number of flights; E_α and $\overline{E}$ are the growth in the proportion of operation cost for Airline α and the expected operation cost for all the airlines respectively. The equation (2) suggests that less difference means more fairness.

2.2 Constraints

There are some constraints which should to be subjected to on ATFNRP.

Constraint 1: $\left[\sum_{f\in F} y_f^{m,n}(\tau,\Delta\tau) \le C_{m,n}(\tau,\Delta\tau)\right] \cap \left[\sum_{f\in F} y_f^{airpt}(\tau,\Delta\tau) \le C_{airpt}(\tau,\Delta\tau)\right]$,

where $y_f^{m,n}(\tau,\Delta\tau) = \begin{cases} 1, \tau_{f,\mathrm{m,n}}^{in} \le (\tau+\Delta\tau) < \tau_{f,\mathrm{m,n}}^{out} \\ 0, otherwise \end{cases}$, $y_f^{airpt}(\tau,\Delta\tau) = \begin{cases} 1, \tau \le \tau_f < \tau+\Delta\tau \\ 0, otherwise \end{cases}$

Constraint 2: $\tau_f \in T_f$, $r_f \in R_f$ and flights between the same original-destination airports pair have the same optional path set

Constraint 3: $\forall r_f \le R, r_f \le (1+50\%)\times \min_{r_f \in R} |r_f|, f \in F$

Constraint 1 limits capacity not only in segments but also in airports (denoted as $C_{airpt}(\tau,\Delta\tau)$) where $\tau_{f,\mathrm{m,n}}^{in}$ and $\tau_{f,\mathrm{m,n}}^{out}$ represent entering and leaving time of $Segment(m,n)$ respectively. The segment capacity restrictions characterize the weather conditions, and the current capacity of $Segment(m,n)$ can be denoted as $W_{m,n}(\tau,\Delta\tau) = \beta \times C_{m,n}(\tau,\Delta\tau)$, where β ($\beta \in [0,1]$) and $C_{m,n}(\tau,\Delta\tau)$ is the capacity under regular weather. Constraint 2 restrains the range of departure time-slots and feasible rerouting paths. Constraint 3 forces the largest extra-distance of feasible rerouting paths.

Compared with the previous multi-objective ATFNRP model, we have taken both segments and airports capacity constraints into consideration and the utilization of ground-holding and rerouting ensure a safe and efficient operation. Hence our model expresses the situations in real-world ATFM better.

3 MOEA/D for Multi-Objective ATFNRP

From section 2 we know that ATFNRP is a multi-objective combinatorial optimization problem essentially which is large-scale with considerable constraints and non-separable decision variables. And, it needs to find the optimal departure time-slots and rerouting paths for all the flights involves in the air traffic flow network. Moreover, the efficiency of solving ATFNRP should be taken into account in real-world circumstance. Therefore, MOEA/D [9] is adapted to address ATFNRP.

In this paper, MOEA/D decomposes the multi-objective ATFNRP into a number of subproblems by using Tchebycheff approach and optimizes them simultaneously, and each optimization of the subproblem is only with the information from its several neighboring subproblems defined based on the distances between their weight vectors. Hence, single objective local search techniques can be readily used in MOEA/D to reduce the computational complexity. In addition, a dynamic penalty function is imposed on MOEA/D to satisfy the large-scale constraints and to accelerate the evolution as well asto search the solution space adequately.

Detailed steps of MOEA/D and the description of the dynamic penalty function and function normalization will be introduced in the following subsections.

3.1 Full Details of MOEA/D

Based on the above, the overall algorithm can be stated as follows:

1. Decomposition: Tchebycheff approach is employed in this paper. Suppose there be N subproblems, and m objectives which is handled by penalty function and normalization described in sector 3.2(m equals to 2 and N is set to 100 in this paper). Each subproblem forms with a weight vector $\lambda^i = (\lambda_1^i, ...\lambda_m^i)^T$, the reference point z^* which is the best value found do far, and the decision variable x, where $x_i = (\mathrm{r}_{1,i}, \mathrm{t}_{1,i}, \mathrm{r}_{2,i}, \mathrm{t}_{2,i}, \cdots, \mathrm{r}_{n,i}, \mathrm{t}_{n,i})$, a $2\times n$ dimensional vector including both the selected departure time-slots and rerouting paths for n flights. The objective function of the i th subproblem can be shown in the following form:

$$g(x|\lambda^i, z^*) = \max_{1\le j\le m}\left\{\lambda_j^i \left| f_j^{new}(x) - z_j^* \right|\right\} \tag{3}$$

2. Initialization: Initialize the N weight vectors as uniform distribution, and compute the distances between any two weigtht vectors to choose T closest as neighborhood to each weight vector.
3. Generation: Generate an initial population $x^1, ...x^N$ randomly and all the departure time-slots and routes are generated within the constraints 2 and 3. Set $gen = 1$ (gen counts the generations of MOEA/D).
4. Recombination: Randomly select two indexes from the neighborhood as parents, and cross them to generate a child solution. Then, a mutation operator is to be

developed to create new solution. The mutation mutates each position of the child solution independently with certain probability.

5. Update of reference point: For each complete solution with departure time-slots and routes, calculate the objectives and choose the minimum values as the new reference point.
6. Update of neighborhood: Compare the newly-obtained solution through recombination with the solutions in the set of its neighborhood, and substitute dominated ones with the new solution.
7. Termination criteria: Once satisfied termination criteria is obtained, output the current solutions and their objective values, and stop the algorithm, otherwise set $gen = gen + 1$ and return Step 4.

3.2 Constraints Handling and Function Normalization

According to the mathematical model presented in Section 2, we know that it is difficult to apply evolution algorithms directly for ATFNRP because it is a multi-objective optimization problem with numerous constraints. Hence, we chose penalty function method [10-12] to handle constraints through considerable methods because penalty function method is one of the most common and effective methods currently. MOEA/D generates the feasible solutions and calculates the objective values without considering the constraints firstly, then it transforms the constraint violation of specific solution into penalty terms and adds them with objective functions. Therefore, multi-objective optimization problem with constraints is transformed to unconstrained one.

Compare to static penalty function method, we adopted dynamic penalty function method to handle constraint 2 which increases as generation grows. The dynamic penalty function is in the form:

$$penalty = \delta(t) \times \{\sum_{s=1}^{S} (\max[0, g_s(x)]) + \sum_{a=1}^{A} (\max[0, g_a(x)])\} \tag{4}$$

Where $\delta(t)$ is the control factor, and is denoted as $\delta(t) = (\frac{gen}{\max\_gen})^2$, and $\max\_gen$ is the total iteration in MOEA/D. Accordingly, with less punishment at the beginning, the algorithm could search a wider range of the space and with larger penalty at later stage, it can concentrate on the feasible region to find better solutions. $g_s(x), g_a(x)$ in equation (4) are the penalty for capacity constraints violation in segment and airport and are defined in equation (5).

$$g_s(x) = {seg\_\max_s}\Big/{(seg\_sum_s + 1)}, g_a(x) = {airpt\_\max_a}\Big/{(airpt\_sum_a + 1)} \tag{5}$$

$$f_j^{'} = \frac{f_j - \min_{f_j}}{\max_{f_j} - \min_{f_j}}, \ penalty^{'} = \frac{penatly - \min_p}{\max_p - \min_p}, f_j^{new} = f_j^{'} + penalty^{'} \tag{6}$$

Where $seg\_\max_s$ and $airpt\_\max_a$ represent the maximum number of flights that exceed the capacity of Segment s and Airport a respectively; seg_sum_s and $airpt\_sum_a$ suggest the total amount of them.

Considering the dimension differences, we should normalize the objective function and penalty function. Normalization is to transform a set of values into a specific interval to eliminate the deviation. Although several methods have been adopted to normalize values, a maximum and minimum standardized method is one of the best-known methods which is chosen in the paper and denoted in equation (6). It is a kind of linear conversion and can retain the relationship exited in the original data.

4 Experimental Studies

The experiments were carried out to analyze the MOEA/D from two aspects: Firstly, the efficacy of MOEA/D was evaluated by comparing with other MOEAs for solving multi-objective ATFNRP. Secondly, the advantages of dynamic penalty function in MOEA/D were analyzed.

The real data extracted from the national airspace of China in 2011 and Flight Schedule Database (FSD) is applied to validate the MOEA/D in our experiments. Fig. 1 shows the corresponding airspace configuration, which comprises 150 airports, 1711 airway segments and 77 sectors. The information of 2112 scheduled flights between the hours of 16:00 and21:00 on Monday contains departure/arrival airport, scheduled departure, aircraft type, etc.

Fig. 1. The air traffic network of China

In our experiment, Sector No.64 was chosen as the capacity-restricted region since it is located in the center and is relatively busy, and all the segments in it were

supposed to be affected. And the weather coefficient β was randomly set to a value in [0,1]. The shortest departure time-interval for each flight of all the airports is set to 1 minute.

4.1 Comparison with Other MOEAs

To assess the efficacy of MOEA/D, the first experiment compares the MOEA/D with some well-known MOEAs including Multi-objective Genetic Algorithm (MOGA) [13], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [14] which has been implemented to solve the similar problem in [7] and Multi-Objective Comprehensive Learning Particle Swarm Optimizer (MOCLPSO) [15] on account of their efficiency for ATFNRP.

Since all the algorithms compared are essentially randomized algorithms, we collected the results of each algorithm according to 10 independent runs and set the number of Fitness Evaluations (FEs) as 643600 for each algorithm so that all algorithms can be fairly compared. The main parameter settings of the compared algorithms are set to the following values suggested in [8, 12]: the population size equals to 100; neighborhood size is 20; the crossover and mutation probability are set to 0.9 and 0.1 respectively.

Fig. 2(a) depicts the performance of the four compared algorithms. The non-dominated solutions obtained by MOEA/D, MOGA, NSGA-II and MOCLPSO in all 10 runs are plotted in the figure. Note that, in terms of both airlines fairness and total delay cost, MOEA/D outperforms the compared three. Moreover, the solutions achieved by MOEA/D dominate all other solutions obtained by the others.

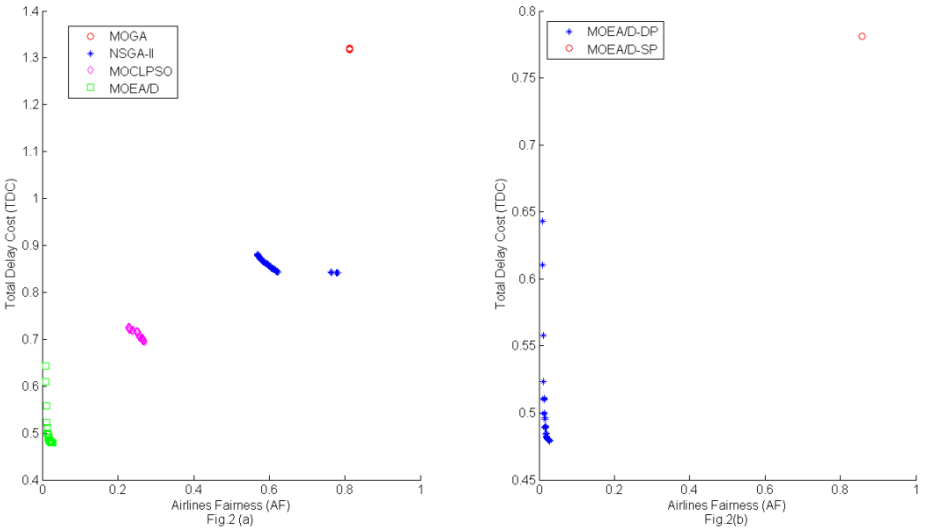


Fig. 2. Pareto front of compared algorithms: Fig.2(a). Non-dominated solutions obtained by all 10 runs of the compared algorithms; Fig.2(b). Non-dominated solutions obtained by MOEA/D with dynamic and static penalty function

To quantitatively evaluate the compared algorithms, the Hypervolume metric (I_H), the distance from reference set (I_D) [16] and the spread (Δ) [14] of the non-dominated solutions were applied to analyze the performance of each algorithm. A better non-dominated set will have higher I_H and lower I_D and Δ. The average values and standard deviations (given in the parenthesis) of I_H, I_D and Δ over the 10 independent runs of each algorithm are presented in Table 1.

For each performance metric, the Wilcoxon rank sum test [17] has been carried out to compare the four methods. In each row of the table, the best value is highlighted in boldface, and it can be noted that MOEA/D outperforms the other three based on I_H and I_D metric while it performs similarly to NSGA-II according to Δ metric.

Table 1. the average value of I_H, I_D and Δ

	MOGA	**NSGA-II**	**MOCLPSO**	**MOEA/D**
I_H	0.8010(0.0220)	0.8677(0.335)	0.9438(0.0396)	**0.9987**(0.0075)
I_D	0.1349(0.0158)	0.0922(0.004)	0.0356(0.0057)	**0.0003**(0.0007)
Δ	1.0930(0.0686)	**0.9991**(0.049)	1.0704(0.0438)	1.0002(0.0264)

4.2 Constraints Handling Test

To verify the contribution of dynamic penalty function, we compare it with static penalty function. The parameter settings are the same as MOEA/D listed above. Fig. 2(b) shows the performances of MOEA/D with dynamic penalty and MOEA/D with static penalty, and table 2 lists the average value of three metrics of them.

Table 2. the average value of I_H, I_D and Δ

	MOEA/D-SP	**MOEA/D-DP**
I_H	0.8095(0.0035)	**0.9999**(0.0056)
I_D	0.1401(0.0032)	**0.0006**(0.0002)
Δ	1.0316(0.0177)	**1.0002**(0.0110)

Note that, the non-dominated set obtained by MOEA/D with dynamic penalty function is better than the static one. Consequently, it demonstrates that dynamic penalty function indeed has significant contribution on constraints handling.

5 Conclusion and Discussion

In this paper, ATFNRP optimizing the airlines fairness and total delay cost simultaneously is investigated. In comparison to previous work, we introduced a multi-objective model for ATFNRP and adopted MOEA/D to seek the optimal departure time-slots and rerouting paths for all flights considered. In the problem formulation,

we took airports capacity constraints into considering, and to handle numerous restrictions, we adopted dynamic penalty function. Comprehensive experiments have been designed and implemented using real data of China airspace and flights plans. The results demonstrated convincingly that, in terms of solution quality, MOEA/D outperforms or performs similarly to three well-acknowledged MOEAs. Although preliminary result is promising, we firmly believe that the model formulation and MOEA/D proposed here can be further improved to get better performance, such as a more comprehensive and detailed model considering more air traffic management actions and a more accurate weather model to improve the practicality of the model.

Acknowledgement. This work was supported by a grant from National Program on Key Basic Research Project of China (973 Program) (Grant No.2011CB707004), and National Science and Technology Support Program of China (Grant No.2012BAG04B01).

References

1. Bertsimas, D., Patterson, S.S.: The Air Traffic Flow Management Problem with Enroute Capacities. Operations Research 46, 406–422 (1998)
2. Bertsimas, D., Patterson, S.S.: The Traffic Flow Management Rerouting Problem in Air Traffic Control: A Dynamic Network Flow Approach. Transportation Science 34(3), 239–255 (2000)
3. Song, K.: The Research of Rerouting Problem in Air Traffic Flow Management. Master thesis, Department of Transportation Planning and Management, Graduate School of Traffic and Transportation Engineering, NUAA, Nanjing (2002)
4. Ye, B.J.: Optimization Model for Tactical Aspects of Air Traffic Flow Management. Master thesis, the Graduate School of Civil Aviation, NUAA, Nanjing (2008)
5. Yaowiwat, S., Lohatepanont, M., Punyabukkana, P.: Multi Objective Micro Genetic Algorithm for Combine and Reroute Problem. International Journal of Intelligent Systems and Technologies 2(4), 245–255 (2007)
6. Bertsimas, D., Lulli, G., Odoni, A.: An Integer Optimization Approach to Large-scale Air Traffic Optimization Flow Management. Operations Research 59(1), 211–227 (2011)
7. Wan, L.L., Tian, Y., Ye, B.J.: The Research of Rerouting Strategy Based on Multi-Objective Evolutionary Algorithm. Mathematics in Practice and Theory 22, 101–108 (2010)
8. Zhang, M., Cai, K.Q., Zhu, Y.B.: An Improved Multi-Objective Particle Swarm Optimization Algorithm For Air Traffic Flow Network Rerouting Problem. In: 31st Digital Avionics Systems Conference, pp. 4B4–4B11 (2012)
9. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
10. Yeniay, Ö.: Penalty Function Methods for Constrained Optimization with Genetic Algorithms. Mathematical and Computational Applications 10(1), 45–56 (2005)
11. Joines, J., Houck, C.: On the Use of Non-stationary Penalty Functions to Solve Non-linear Constrained Optimization Problems with Gas. In: Proceedings of the First IEEE International Conference on Evolutionary Computation, pp. 579–584. IEEE Press (1994)

12. Kazarlis, S., Petridis, V.: Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase of the Dynamic Penalty Terms. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 211–220. Springer, Heidelberg (1998)
13. Daniel, D., Oussedik, S., Stephane, P.: Airspace Congestion Smoothing by Multi-objective Genetic Algorithm. In: ACM Symposium on Applied Computing, pp. 907–912 (2005)
14. Deb, K., Agarwal, S., Pratap, A., et al.: A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computing 6(2), 182–197 (2002)
15. Huang, V.L., Suganthan, P.N., Liang, J.J.: Comprehensive Learning Particle Swarm Optimizer for Solving Multiobjective Optimization Problems. International Journal of Intelligent System 21(2), 209–226 (2006)
16. Czyzzak, P., Jaszkiewicz, A.: Pareto Simulated Annealing – A Metaheuristic Technique for Multiple-objective Combinatorial Optimization. Journal of Multi-Criteria Decision Analysis 7(1), 34–47 (1998)
17. Wilcoxon, F.: Individual Comparisons by Ranking Methods. Biometrics Bulletin 1(6), 80–83 (1945)

Temporal Dependence in Legal Documents*

Daniel Isemann, Khurshid Ahmad, Tim Fernando, and Carl Vogel

School of Computer Science and Statistics, Trinity College Dublin, Ireland
{isemandi,kahmad,tfernand,vogel}@tcd.ie

Abstract. Tasks and difficulties inherent in the largely open problem of temporal information extraction from legal text are outlined. We demonstrate the efficacy of tools and concepts available "off-the-shelf" and suggest refinements for such applications. In particular, the frequent references between regulatory texts have to be addressed as a separate named entity recognition task that bears relevance to an analysis of the temporal ordering of legislation. A regular expression-based approach as a robust first step towards addressing this problem is tested.

Keywords: named entities, temporality, legal information extraction.

1 Introduction

Information on time and events, their ordering and causal dependencies, is critical in the legal domain and in particular in financial industry regulation. In a given regulatory framework certain requirements may have to be met by a certain deadline or particular disclosures can only be made within a certain time after an event in order to be in compliance. Such time dependence has motivated researchers in compliance information systems to experiment with and recommend the use of temporal logics [1–4].

Documents in a given regulatory domain, say capital adequacy or anti-money laundering, contain the names of key objects and events. It is not difficult for human readers to extract information about the duration of an event or the changing nature of legislation, often by analysing the tense information in verbs. A handle on dates of origination of legal documents gives a temporal sequence from earliest to latest laws which may indicate time-spans of a law's validity.

For temporal analysis we rely on in-text evidence with a view to automating the extraction of relevant information. We present an analysis of whether existing NLP systems can deal with legal and regulatory framework mnemonics which contain dates and references to jurisdictions, and following work elsewhere [5], we find regular expressions well-suited. In §2 we will make the problem statement more precise and review relevant research contributions. In §3 we report on existing off-the shelf tools and characterize their strengths and limitations for the given tasks in our domain. In §4 we motivate an alternative approach to extracting and managing the temporal interconnectedness of legislation and outline a first implementation which we have undertaken. We conclude in §5.

* We gratefully acknowledge Enterprise Ireland's *Governance Risk and Compliance Technology Centre* Initial Grant CC-2011-2601-B.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 497–504, 2013.

2 Named Entity and Date Extraction in Legal Text

Legal draughting introduces named entities in legal texts in a variety of ways. Next to references to people, organisations, places, events and objects, conventional legal text contains references to other laws which may either be in force, have been repealed or may be forthcoming. References to other items of legislation and to judicial (court judgements) and executive documents (enforcement documents, regulations based on legislation), rely on a subtle form of abbreviation which forms mnemonics for existing laws together with dates of their origination, e.g. *UK MLR 2007* or *EU Directive 2005/29/EC*.

Named entity recognition (NER) is often taken to be the starting point of information extraction and a first step in relation detection. Similarly, event detection has been suggested as a first step in temporal analysis. Standard definitions of these tasks, however, reveal how much default assumptions are shaped by the prototypical domains of newswire text and news reportage: "Generic NER systems tend to focus on finding the names of people, places and organizations that are mentioned in ordinary news texts" [6, p. 760]. Typical subtasks of temporal analysis are "Fixing the temporal expression with respect to an anchoring date or time, typically the dateline of the story in the case of news stories; [...] Arranging the events into a complete and coherent timeline." [ibid., p. 761]. While we believe that temporal information plays an important role in legislative documents, these documents, unlike news text do not exhibit a narrative structure. In most cases there is no "story" and no "complete and coherent timeline". Instead, as we will argue in §3, we believe that there are several interacting timelines which need to be taken into account. As relationships between entities mentioned in legal documents, may involve references to time, we take named entity recognition as a precursor to temporal analysis.

Several researchers have addressed the task of named entity recognition in legal texts. Dozier and colleagues have applied a hybrid method of controlled vocabulary lookup, contextual rules and statistical models to a test set of 600 US trial documents to identify named entities such as jurisdiction, court, document title, document type and name of the judge [7]. They report high precision for identifying judges (98%) and high recall for capturing jurisdictions (87%). Quaresma and Gonçalves use linguistic information to identify named entities in a corpus of legal documents from the International Agreements/External Relations section of the EUR-Lex portal[1] and classify documents based on some of the named entities discovered [8]. Unlike Dozier et al., Quaresma and Gonçalves found only very few person names in their corpus (presumably due to nature of international agreements). For the NER part the authors focus on the categories of location and organization names, dates and references to documents and document articles. We are not aware of work which systematically addresses the identification of statutes and relationships between legal documents and specific dates.

[1] `eur-lex.europa.eu/en/index.htm` (last accessed 20/07/2013).

3 Existing Linguistic Mark-Up Tools

We present an analysis of how the extraction of time and event related information may be supported by more traditional attempts at automated text analysis (such as tagging and named entity recognition) in making sense of legal documents. We have experimented on a sample text of UK anti-money laundering (AML) regulatory guidelines. For this we hand-annotated the occurrence of named entities in a small sample of text from a regulatory authority which we then subjected to automated analysis for named entity recognition and temporal analysis in this domain with readily available domain independent natural language processing tools. We outline limits of such non-customized approaches. We use a qualitative rather than quantitative analysis to highlight where standard methods fall short and in turn motivate our own approach (§4). This is in the spirit of recall analysis of precision grammars applied to large corpora [9].

We hand-annotated named entities in two paragraphs of an HM Revenues & Customs document on anti-money laundering guidelines for money service businesses [10] and applied the Stanford University Tagger, the Stanford CoreNLP NER module[2] and the TARSQI toolkit for temporal mark-up[3] to this text.

3.1 Off-the-Shelf Named Entity Recognition in the GRC Domain

Studying this text segment we identified four different types of named entities. Three of these are well known candidates from generic named entity recognition tasks: date, location and organization. The fourth category consists of references to other legislative and regulatory documents.[4] We believe that the last category, references to other regulatory documents is particularly important in our domain and that it can be leveraged for a temporal analysis as well. The consistent identification of this category is largely an open problem. While Dozier et al. mention the need to identify such references[5] they do not attempt it in their analysis [7]. Quaresma and Gonçalves, on the other hand, emphasize the importance of this category: "we can state that these documents have a high number of references to other documents and articles [...] this kind of information [...] would allow the inference of important relations, such as, the chain of legislation references." [8, p. 55]. However, they report an error rate of 65% for their effort to automatically identify such references from an analysis of parse trees [ibid.].

The following sentence taken from our sample illustrates three of the four different types of named entities we distinguished:

> *In addition, under the Money Laundering Regulations 2007*^{REF}*, which came into force on 15 December 2007*^{DATE}*, HMRC*^{ORG} *will have powers to cancel the registration of Money Transmission Businesses*^{ORG} *where they are found to be consistently non-compliant with the Payments Regulation*^{REF}*.*

[2] Available at `nlp.stanford.edu:8080/corenlp` (last accessed 20/07/2013).

[3] Available at `www.timeml.org/site/tarsqi/toolkit/index.html` (last accessed 20/07/2013).

[4] Note that our categories are identical to the ones studied by Quaresma et al. [8].

[5] We take their example of identifying "statues [sic]" [7, p. 28] to be intended to refer to statutes.

Table 1 gives an overview of the four categories, their definition and how often they occurred in our exploratory text sample. There were 22 named entities.

Consider a set of five sentences taken form this document together with the tags attributed to the words and sometimes phrases by the Stanford Tagger and NER module (Table 2). We have only shown tags that describe location names, organization names, dates, (other) named entities and proper nouns. Phrases tagged with these tags have been underlined. Note that in some instances the tagger fails to recognize a proper noun, like 'Transfer Regulation' in Payments/Wire Transfer Regulation (sentence #1) but in other cases the NER tags a complex phrase, say Money Laundering Regulations (sentence #3), accurately as a named entity. The NER module recognizes the EC and Her Majesty's Revenue and Customs (HMRC) as an organization. The recognition of dates and organization names requires improvement insofar as dates and organisations that are part of legal or regulatory references are recognized as separate entities.

Using our hand-annotated standard as a reference we computed the Stanford system's precision (52%), recall (50%) and F1 score (51%). While the achieved F1 score still leaves room for improvement, it has to be recalled that named entity recognition is known to be a highly domain specific task. Common types of named entities, such as dates, can be identified very reliably. Domain specific categories, such as references to laws and regulations, however, (our REF category, Table 1), cannot be identified by off-the-shelf tools. Generic entity recognition therefore has to be supplemented with algorithms specific to the legal domain.

From our analysis we noted that references to regulatory documents or parts thereof, which frequently occur in legal text, are partially identified by existing parsers as proper nouns. However, there are two drawbacks of a parser-based analysis: First, fine-tuning is necessary to correctly categorize the type of such entities as references and to adequately delineate these references as they often include dates and organization names. Second, parsers and NER systems usually make no attempt to link dates and other temporal information to mentions of events or to embed these into a wider temporal context. In order to address this we have turned to TimeML and the associated Tarsqi system.

Table 1. The categories of named entities that can be found in regulatory documents and that are targeted through text analysis

Category	Definition	Number
DATE	Descriptions of dates, including days, months, years or combinations thereof in text.	3
LOC	Geographical location names.	1
ORG	Organisation names.	3
REF	References to legislative or regulatory documents including subsections or appendices of documents.	15

Table 2. Sentences from a UK HMRC document on anti-money laundering [10] showing selected entities that are recognized by the Stanford Tagger and NER application

1. The EC^{ORG} Payments/Wire$^{PROPER\ NOUN}$ Transfer Regulation came into effect on the 1st January 2007^{DATE}.
2. The UK^{LOC}'s supervision and enforcement provisions are set out in the Transfer of Funds (Information on the Payer) Regulations$^{PROPER\ NOUN}$ 2007^{DATE}.
3. In addition, under the Money Laundering Regulations$^{OTHER\ ENTITY}$ 2007^{DATE}, which came into force on 15 December 2007^{DATE}, HMRC^{ORG} will have powers to cancel the registration of Money Transmission$^{OTHER\ ENTITY}$ Businesses where they are found to be consistently non-compliant with the Payments Regulation$^{OTHER\ ENTITY}$.
4. For more information about HMRC^{ORG}'s powers to cancel registrations, please refer to MLR9 Registration notice.
5. Both the Money Laundering Regulations$^{PROPER\ NOUN}$ 2007^{DATE} and EC^{ORG} Regulation$^{PROPER\ NOUN}$ 1781/2006 on information on the payer accompanying transfers of funds (Payments Regulation/Wire Transfer Regulation$^{OTHER\ ENTITY}$) require verification of customers' identities 'on the basis of documents, data or information obtained from a reliable and independent source'.

3.2 Off-the-Shelf Temporal Mark-Up in the GRC Domain

Document mark-up has recently progressed from marking up typographical and display related information to marking up time and event related information in text. One such standard is *TimeML* [11]. There are several programs to identify such information automatically through the use of natural language processing technology [12, 13]. As a representative of such programs we have chosen the *TARSQI* toolkit, an experimental mark-up system for TimeML available under a free of charge license [12]. We have used TimeML and TARSQI for a qualitative analysis of the incidence of time related information in GRC documents.

We ran TARSQI on our sample text [10] and studied the temporal links introduced by the system. In total, TARSQI suggested 16 links between events and/or temporal anchors in the text. Twelve of these were temporal links, comprising eight BEFORE and four SIMULTANEOUS links.[6] Insofar as TARSQI identified dates correctly, it had no difficulty temporally ordering them. Some of the system guesses were inaccurate, however. In "EC Regulation 1781/2006", TARSQI identified "1781" as a year. A different picture emerged for the temporal ordering of events. Several of the links that TARSQI identified between postulated events appeared "broken" due to an incommensurability of the timelines involved.

In linguistics literature a distinction is sometimes made between *episodic* and *generic* statements or clauses [14]. We believe this distinction to be of importance in the context of temporal links. An episodic statement is one that describes one or more actual, concrete events. Generic statements on the other hand describe general truths, laws, rules or expectations. An example of an episodic statement from our text on anti-money laundering is: "The EC Payments/Wire Transfer

[6] Note that the TARSQI toolkit attempts to guess temporal as well as non-temporal TimeML links, such as modal or counter-factive subordination or aspectual links.

Regulation came into effect on the 1st January 2007." An instance of a generic statement from the same document is: "Where there is a higher risk of false identity documents or information, there may be a need to obtain additional evidence of identity." These two kinds of statements may have to be treated separately as far as temporal links (e.g. BEFORE, SIMULTANEOUS etc.) are concerned because links crossing from the episodic to the generic realm and vice versa are likely to span different, not directly comparable notions of time and timelines. In our sample paragraphs we found that one in five sentences was of episodic, the others of generic nature. Further research is needed to quantify which ratio of generic to episodic clauses is typical of regulatory documents and to establish which textual markers (e.g. mentions of concrete dates) can be used to algorithmically distinguish between the two kinds of statements.

4 Temporal Analysis of Legislative Documents

The importance of temporal information in legal texts is underlined by the advanced search facility of the EUR-Lex portal which offers no less than 14 relations between a legal document and a date for selection, including: *Date_of_publication*, *Date_of_effect*, *End_of_validity_date*, *Date_of_signature*, *Date_of_debate*, etc.[7] Relationships between documents may also have temporal implications. Most of the 17 relationships between documents listed by the EUR-Lex advanced search cover some sort of amendment or legal effect that presupposes a before and after relation between legal documents and sometimes may affect the validity of legislation. We analysed 70 EU directives from the Wolters Kluwer Compliance Resource network[8] for dates and other directives mentioned. Using regular expressions we robustly identified the occurrences of dates and references to other EU directives. On a sample of seven randomly selected EU directives, our directive pattern, which had perfect precision, achieved 86.4% recall.

With such patterns we analyzed the EU directives for cross-references and dates and made an attempt at qualifying references as legal amendments (Figure 1). We found that references to other directives though widespread throughout the documents are typically focussed on a small number of key directives per document. While on average a EU directive references 16.1 other directives (SD = 9.7), it is usually less than four directives (3.6, SD = 2.7) that account for more than half of the individual reference count in a given document.

Studying the cross-references we found that some directives receive many more incoming links than others, perhaps indicating that these are of foundational character.[9] Qualifying a reference as legally amending another one, one can begin to create timelines of laws that wholly or in part amend each other as a first step towards establishing time-spans of legislation validity.

[7] `eur-lex.europa.eu/expert/sg/sga_cnct/celexexp!dev?LANG=EN&BASE=bas-cen` (last accessed 20/07/2013). One of the 14 relations is a super-category (*All_dates*).

[8] `www.complianceresourcenetwork.com/web/crn` (last accessed 20/07/2013).

[9] See `www.scss.tcd.ie/~kahmad/IDEAL2013/` for the data set, a sample cross-reference graph and the patterns discussed here.

Directive ID	Directive Title	Publication Date	Amended Directives	Referenced Directives	Dates mentioned
Directive 2004/39/EC	Oganisational requirements and operating conditions for investment firms - Draft implementing Directive 2004/39/EC	Fri Jun 02 00:00:00 IST 2006	[]	[Directive 85/611/EEC x 8, Directive 2003/125/EC x 6, Directive 2003/6/EC x 5, Directive 2000/12/EC x 4, Directive 2003/71/EC x 4, Directive 93/22/EEC x 1, Directive 2005/29/EC x 1, Directive 2002/65/EC x 1, Directive 95/46/EC x 1, Directive 84/450/EEC x 1]	[Sat Nov 01 00:00:00 GMT 2008 x 1, Tue Nov 04 00:00:00 GMT 2003 x 1, Mon Dec 22 00:00:00 GMT 2003 x 1, Wed Apr 21 00:00:00 IST 2004 x 1, Wed Jan 31 00:00:00 GMT 2007 x 1, Thu Nov 01 00:00:00 GMT 2007 x 1, Fri Oct 31 00:00:00 GMT 2008 x 1, Fri Dec 20 00:00:00 GMT 1985 x 1]
Directive 90/211/EEC	Mutual recognition of public-offer prospectuses as stock-exchange listing particulars Directive 90/211/EEC	Mon Apr 23 00:00:00 IST 1990	[Directive 80/390/EEC x 1]	[Directive 80/390/EEC x 4, Directive 89/298/EEC x 2, Directive 87/345/EEC x 1]	[Mon Apr 23 00:00:00 IST 1990 x 2, Wed Apr 17 00:00:00 IST 1991 x 1]

Fig. 1. Pattern-based information extraction from EU Directives: Example output

5 Concluding Remarks

The analysis with tailored regular expressions for statutes and relations among dates and statutes as named entities in §4, along with cross-reference assessment involves computationally simpler technology than the more sophisticated tools considered in §3.1 and §3.2, but the methods are well suited to the purpose.

We outlined some of the tasks and difficulties inherent in the largely open problem of temporal information extraction from legal text. Two important aspects of documents in this domain were identified. First, the frequent references from one regulatory text to another have to be addressed as a separate named entity recognition task. We have motivated the relevance of this task to an analysis of the temporal ordering of legislation. A pattern-based approach provides a robust first step towards a solution. Second, in terms of the identification and interpretation of temporal links within a given regulatory text, a distinction between episodic and generic statements has to be made and care has to be taken that temporal links do not span this divide.

Several lines of future work arise from our preliminary study. First, we plan to combine our pattern-based approach for reference and link extraction with suitable machine learning techniques, hopefully enhancing and generalizing our extraction results. Second, we plan to build a classifier for telling episodic from generic statements, a useful first step for analyzing in-document temporal relations, as we have argued. Further exploring the linkages between legal documents one may create a page-rank hierarchy on a corpus of such documents, similar to those known from information retrieval. Such a hierarchy can be embellished further by analyzing the distribution of the frequency of keywords in each of the documents in the corpus and we expect semantic annotation frameworks such as RDF to be useful in this context. This approach may benefit professional providers of legal content, such as Wolters Kluwer or Thomson Reuters. Finally, the fact that in the context of European legislation, each EU member state may have a different way of interpreting a directive or regulation may necessitate a multi-lingual analysis that has to deal with local exemptions and derogations.

References

1. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. ACM Transactions on Information Systems (TOIS) 8(4), 325–362 (1990)
2. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1), 3–50 (1993)
3. Fuxman, A., Liu, L., Mylopoulos, J., Pistore, M., Roveri, M., Traverso, P.: Specifying and analyzing early requirements in Tropos. Requirements Engineering 9(2), 132–150 (2004)
4. COMPAS-Project: Deliverable D2.2: Initial specification of compliance language constructs and operators, Version 2.0 (2009), `www.compas-ict.eu/compas_results/deliverables/m11/D2.2_Initial-specification-of-compliance-language-constructs-and-operators.pdf` (last accessed July 20, 2013) (last accessed July 20, 2013)
5. Breaux, T.D., Gordon, D.G.: Regulatory Requirements Traceability and Analysis Using Semi-formal Specifications. In: Doerr, J., Opdahl, A.L. (eds.) REFSQ 2013. LNCS, vol. 7830, pp. 141–157. Springer, Heidelberg (2013)
6. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice Hall Series in Artificial Intelligence. Pearson Education International, New Jersey (2009)
7. Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., Wudali, R.: Named Entity Recognition and Resolution in Legal Text. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.) Semantic Processing of Legal Texts. LNCS (LNAI), vol. 6036, pp. 27–43. Springer, Heidelberg (2010)
8. Quaresma, P., Gonçalves, T.: Using linguistic information and machine learning techniques to identify entities from juridical documents. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.) Semantic Processing of Legal Texts. LNCS (LNAI), vol. 6036, pp. 44–59. Springer, Heidelberg (2010)
9. Baldwin, T., Beavers, J., Bender, E., Flickinger, D., Kim, A., Oepen, S.: Beauty and the Beast: What Running a Broad-Coverage Precision Grammar over the BNC Taught Us about the Grammar – and the Corpus. In: Kepsar, S., Reis, M. (eds.) Linguistic Evidence: Empirical, Theoretical and Computational Perspectives. Studies in Generative Grammar, vol. 85, pp. 49–69. Mouton De Gruyter (2005)
10. HM Revenue & Customs: Anti-money laundering guidance for money service businesses, We used paragraphs 10.5.6 and 10.6.1 for the analysis described in this paper (2010), `www.hmrc.gov.uk/mlr/mlr_msb.pdf` (last accessed July 20, 2013)
11. Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: Fifth International Workshop on Computational Semantics (IWCS-5) (2003)
12. Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S.B., Littman, J., Rumshisky, A., Phillips, J., Pustejovsky, J.: Automating Temporal Annotation with TARSQI. Demo Session. In: Proceedings of the ACL (2005)
13. Llorens, H., Saquete, E., Navarro, B.: Tipsem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, pp. 284–291 (2010)
14. Carlson, G., Pelletier, F.: The Generic Book. University of Chicago Press (1995)

Learning-Guided Exploration in Airfoil Optimization

Edgar Reehuis[1,2], Markus Olhofer[2], Bernhard Sendhoff[2], and Thomas Bäck[1]

[1] Natural Computing Group, LIACS, Leiden University
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
{ereehuis,baeck}@liacs.nl
http://natcomp.liacs.nl
[2] Honda Research Institute Europe GmbH
Carl-Legien-Straße 30, 63073 Offenbach/Main, Germany
{markus.olhofer,bernhard.sendhoff}@honda-ri.de
http://www.honda-ri.de

Abstract. A learning-based exploration approach is proposed to escape from the basins of attraction of converged-to optima, by selecting on what is termed the interestingness of a solution. This interestingness is based on the modeling error made by a surrogate model that is trained on all solutions encountered earlier during the search. Compared to multiple standard optimization runs, a learning-guided restart scheme that alternates between a quality optimization phase and an exploration phase directed by interestingness finds solutions that are more diverse and of higher quality.

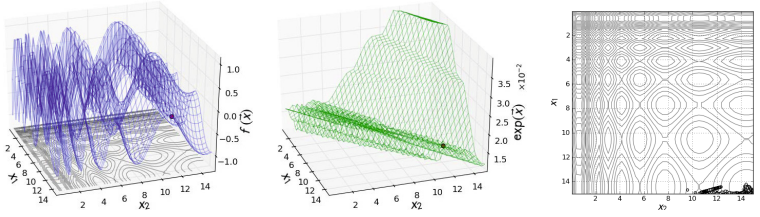
Keywords: Novelty, interestingness, diversity, restarts, airfoil optimization.

1 Introduction

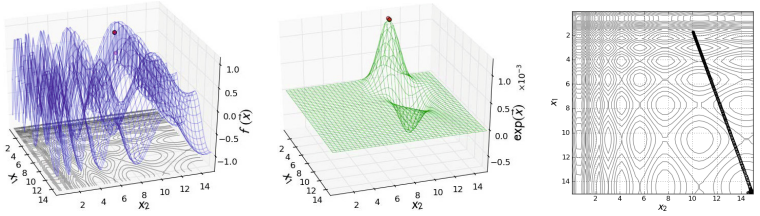
In airfoil optimization it can be necessary to start optimization from a basic initial solution. This is required when it is not straightforward to generate a random initial solution, for instance when simulation of aerodynamic behavior only converges for solutions that meet specific geometric constraints. Different runs of a stochastic search approach may then result in similar optimized solutions.

In order to escape the "default" basin of attraction we propose using an additional search objective specifically aimed at exploration. An overview of possible measures that derive from the history of solutions encountered earlier during the search is given in [4], making a division between similarity-derived, *distance-based* measures and *learning-based* measures related to fitting a *surrogate model*. A scheme that alternates between a quality optimization phase and a distance-based exploration phase is described in [5], while in this work we present a similar scheme using a learning-based guide, both applied to airfoil optimization. The surrogate model is *not* intended here for reducing the number of used quality function evaluations, but for indicating the modeling error and thereby the *understanding* in a certain region during optimization.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 505–512, 2013.



(a) Fitness Function (b) **Novelty** (c) Sampled Points



(d) Fitness Function (e) **Interestingness** (f) Sampled Points

Fig. 1. Novelty vs. Interestingness. Sampling behavior displayed for a (3,6)-CMA-ES [2] selecting on novelty vs. its sampling behavior in selecting on interestingness. Selecting on novelty causes stagnation (see [4]), while in selecting on interestingness newly sampled solutions get high scores that degrade to negative interestingness over time, thereby sustaining exploration.

Following, Section 2 lays out the learning-based exploration measure and how it is integrated as exploration criterion in the search, after which the results of applying the resulting scheme to an airfoil optimization task are given in Section 3. We conclude with discussion and outlook in Section 4.

2 Novelty and Interestingness

Learning-based novelty is expressed as the error that a surrogate model makes in approximating the mapping of solution vectors to, e.g., quality-function values. In selecting on learning-based novelty, solutions with maximum error are promoted. However, the search can get stuck in areas for which the model's approximation does not improve over time (see Fig. 1). To steer away from regions that effectively cannot be modeled, we should select the solution that induces maximum *learning progress* instead of the one with maximum error. Learning progress induced by a solution is the observed improvement in approximating the to-be-learned mapping by including the solution in model training. It is estimated by the *interestingness* of a solution, which is derived from the solution's novelty value as well as the novelty values of earlier sampled solutions (see Fig. 2).

Dispersion in Predictions. For expressing learning-based novelty we use *dispersion in predictions* (DP) [4]. Multiple models are trained on the same data,

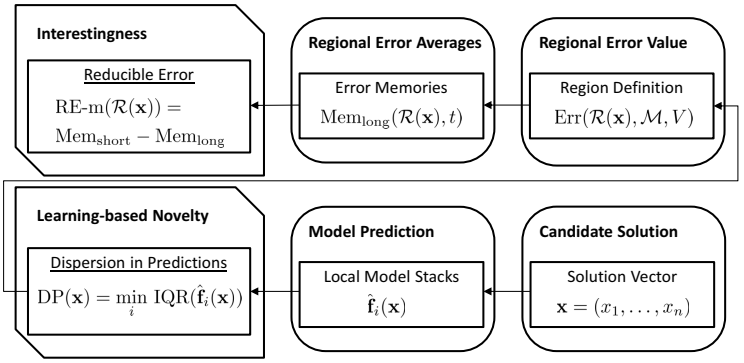


Fig. 2. Calculation of Interestingness Expression RE-m (DP). Starting from the *learning-based novelty* of solutions at their time of sampling, the influence of these earlier observed error values is calculated (*regional error value*) depending on their proximity to the point for which interestingness is to be evaluated. An interestingness value is then obtained based on how the error in the region in which the solution lies has changed over time (*regional error averages*).

and the dispersion between their predictions serves as estimation of the modeling error. DP requires surrogate models with potentially differing outcomes in presence of limited training data. We calculate DP as the *interquartile range* (IQR) of the quality predictions for a solution $\mathbf{x}$ by the $\dim(\hat{\mathbf{f}})$ stacked models,

$$\text{DP}(\mathbf{x}) = \text{IQR}\left(\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_{\dim(\hat{\mathbf{f}})}(\mathbf{x})\right). \tag{1}$$

To be prepared for modeling irregular quality landscapes, we use multiple *local model stacks* that together emulate a global model stack (see Fig. 3). The different local model stacks are trained on disjoint or partly shared data. DP is then calculated as the minimum IQR score resulting from evaluating a solution $\mathbf{x}$ on all local model stacks,

$$\text{DP}(\mathbf{x}) = \min_i \ \text{IQR}\left(\hat{\mathbf{f}}_i(\mathbf{x})\right). \tag{2}$$

Reducible Error. Using the global model stack $\mathcal{M}(T_t)$ we obtain from observed DP-values a *regional error value* for the region $\mathcal{R}(\mathbf{x})$ in which solution $\mathbf{x}$ lies [4],

$$\text{Err}(\mathcal{R}(\mathbf{x}), \mathcal{M}(T_t), V_t) = \max_{\mathbf{v} \in V_t} \left(\underbrace{\text{DP}(\mathbf{v})}_{\text{on } \mathcal{M}(T_t)} \cdot \underbrace{(2\pi)^{-\frac{1}{2}|\mathbf{v}|} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\mathbf{v})'\mathbf{I}(\mathbf{x}-\mathbf{v})\right)}_{\text{pdf}_{\mathcal{N}(\mathbf{v},\mathbf{I})}(\mathbf{x})} \right), \tag{3}$$

that is, we determine test point $\mathbf{v}$ with the highest resulting DP-value after decreasing it relative to the distance between $\mathbf{x}$ and $\mathbf{v}$, based on the probability

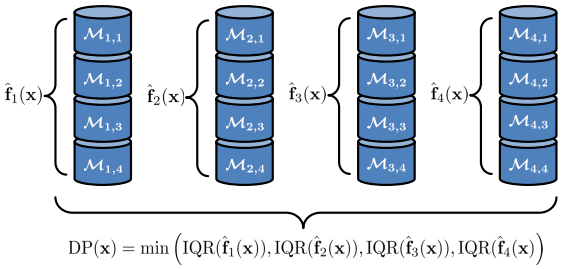


Fig. 3. Local Model Stacks. A global model stack is obtained using an ensemble of local model stacks. Within local stack i all surrogate models $\mathcal{M}_{i,j}$ are trained on the same training data, while different local stacks use data from different areas in the search space. The lowest DP score is selected after evaluating a solution $\mathbf{x}$ on all stacks.

density function of a normal distribution around $\mathbf{v}$. The global model stack is trained on the set T_t of all solutions evaluated before time step t, and test set V_t contains all candidate solutions $\mathbf{x}$ at time step t. If no training data is available at time t, $\mathrm{Err}(\mathcal{R}(\mathbf{x}), \mathcal{M}(T_t = \emptyset), V_t)$ is set to 0.

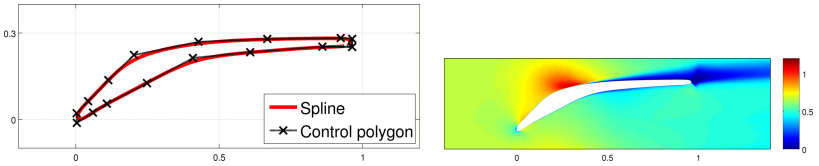
From the regional error values we obtain *regional error averages* at different points in time, in the form of a *short-term* and *long-term memory*. We express interestingness by assuming that the learning progress will be maximized in a region with high *current* error, as the model is still able to greatly improve there, but account for regions that cannot be modeled by subtracting the *irreducible* error, that is, the error in the region at an earlier time step. This results in the *reducible error* (RE) [4], proposed by Luciw et al. [3],

$$\text{RE-m}(\mathcal{R}(\mathbf{x})) \;=\; \underbrace{\mathrm{Mem}_{\mathrm{short}}(\mathcal{R}(\mathbf{x}), t)}_{\underline{\text{current}}\text{ error in }\mathcal{R}(\mathbf{x})} \;-\; \underbrace{\mathrm{Mem}_{\mathrm{long}}(\mathcal{R}(\mathbf{x}), t)}_{\underline{\text{irreducible}}\text{ error in }\mathcal{R}(\mathbf{x})}\,, \tag{4}$$

using *exponential smoothing* with weighting for the memories as proposed in [3],

$$\begin{aligned}\mathrm{Mem}_{\mathrm{short}}(\mathcal{R}, t) &= (1 - 0.2) \cdot \mathrm{Mem}_{\mathrm{short}}(\mathcal{R}, t-1) + 0.2 \cdot \mathrm{Err}(\mathcal{R}, \mathcal{M}(T_t), V_t),\\ \mathrm{Mem}_{\mathrm{long}}(\mathcal{R}, t) &= (1 - 0.1) \cdot \mathrm{Mem}_{\mathrm{long}}(\mathcal{R}, t-1) + 0.1 \cdot \mathrm{Err}(\mathcal{R}, \mathcal{M}(T_t), V_t).\end{aligned} \tag{5}$$

Learning-Guided Restart Scheme. To apply interestingness as exploration criterion, we take inspiration by Cuccu et al. [1] and use a restart scheme that alternates between a fitness optimization phase and an exploration phase, originally proposed for distance-based exploration in [5]. The alternating scheme is aimed at escaping basins of attraction of earlier converged-to optima. For the considered airfoil optimization, the search space consists largely of areas with *infeasible* solutions (i.e., violating some geometric constraint, or for which the performance cannot be simulated). We perform optimization on fitness until some convergence criterion is met, and then optimize on interestingness until all solutions in the current population have become infeasible. Following, the fitness optimization phase is restarted, using a variable penalty for infeasible



(a) Spline Representation (b) Mach Number Distribution

Fig. 4. Initial Geometry. In (a) the 16 *control points* and their *non-uniform rational B-spline* (NURBS) decoded geometry is shown. In (b) the *Mach number distribution* (the relative air speed divided by the local speed of sound) of the simulated *flow field* around the geometry is displayed, showing a deviating outflow for the initial geometry.

solutions depending on the degree of infeasibility to be able to find a way back to feasible space. The alternating scheme is run on top of the *CMA-ES* [2]: Between restarts its covariance matrix is reinitialized and its stepsize reset to the, problem-specific, initial value.

3 Experiments

Two-dimensional optimization is performed on the *outlet guide vanes* (OGVs) for the *low-pressure compressor* in a small turbofan engine, with 16 control points for adjusting a geometry, thus 32 decision variables (see Fig. 4). OGVs are *stator blades*, non-moving blades intended for straightening the air flow, arranged around a central axis in a cylindrical fashion. Simulating aerodynamic behavior is done using an in-house deterministic *Navier-Stokes flow solver*.

The learning-guided restart scheme is run on top of a (12,24)-CMA-ES for 5000 generations (120k evaluations by the simulator). It is compared against multiple standard optimization runs using the same (12,24)-CMA-ES for 1000 generations each (24k evaluations). A problem-specific initial stepsize σ_{init} of 1×10^{-3} is used.

Within the learning-guided restart scheme we stop the fitness optimization phase when the stepsize is smaller than $0.5 \times \sigma_{\text{init}}$. In the exploration phase the stopping condition is that either all candidates have become strictly infeasible (a geometry is disconnected, i.e., contains loops, or the simulator does not converge for it), or the search is in a maximally erratic area for still allowing to find a way back to a high-performing solution (the average fitness value of the best individual over the last 5 generations exceeds 1×10^3).

A local model stack consists of 10 *feed-forward neural networks* (FFNN) and each generation a new stack is trained on the last 3 generations of evaluated solutions. An ensemble of disjoined local stacks is maintained by keeping the local model stack of each third generation. The FFNN have linear outputs and a hidden layer of 25 sigmoidally-activated nodes, a bias node, and are trained using improved Rprop [4] for a maximum of 10k epochs from initial connections weights in $[-0.1, 0.1]$. They are fully connected, including direct connections from input to output nodes. All data is used for training, no validation set is

used. The CMA-ES and learning-guided restart scheme have been implemented using the Shark Machine Learning Library v2.3.43.

The target of the search is twofold: Finding a single solution of *maximal* quality and producing a set of *diverse* solutions with *acceptable* quality. To omit a priori specifying a level of acceptable quality, we take the quality value of each solution in a method's result set as possible minimal level. This gives rise to multiple alternative *level set* approximations within the result set for which we can determine the diversity. Each such "solution level set" contains the solution itself and all other found solutions that have quality equal to or better than the solution's quality score, and this way we associate a diversity score with each solution. From the resulting quality/diversity pairs a two-dimensional *Pareto front* is obtained that illustrates the performance of the method.

Fitness Function. The quality of a solution is based on seven properties that are aggregated into a to-be-minimized fitness value: The resulting deviation angle of the air flow, the coefficient of the occurring pressure loss, the extent of violation of four minimal thickness constraints along the blade, and a penalty depending on the extent to which the solver has not converged.

Diversity Calculation. We use *Solow-Polasky* [5] to express diversity in a set of solutions S by applying *domain-specific distance measure* $\mathrm{d}_{\mathrm{airfoil}}$ [5],

$$\mathrm{D}_{\mathrm{SP}}(S) = \sum_{i,j} b_{ij}, \tag{6}$$

with $\boldsymbol{B} = \boldsymbol{A}^{-1}$, $a_{ij} = \exp(-\mathrm{d}_{\mathrm{airfoil}}(\mathbf{x}_i, \mathbf{x}_j))$, and $1 \leq i, j \leq |S|$, that is, calculate the sum of the elements b_{ij} of $\boldsymbol{B}$, the inverse of $\boldsymbol{A}$. The elements a_{ij} of $\boldsymbol{A}$ depend on the difference between all pairs of solutions $\mathbf{x}_i, \mathbf{x}_j$ from set S. $\mathrm{D}_{\mathrm{SP}}(S)$ results in a real number in the interval $[1, |S|]$. For calculating $\mathrm{d}_{\mathrm{airfoil}}$ a geometry $P(\mathbf{x})$ is sampled in 200 points equidistantly along the spline. Each sample is divided in a *suction* side (side featuring higher air speed, upper side in Fig. 4b) set and *pressure* side (side featuring lower air speed, lower side in Fig. 4b) set, and the *summed-up Hausdorff distance* $\mathrm{d}_{\Sigma\mathrm{H}}$ [5] between the respective sets is added up,

$$\mathrm{d}_{\mathrm{airfoil}}(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{d}_{\Sigma\mathrm{H}}(\mathrm{s}(P(\mathbf{x}_1)), \mathrm{s}(P(\mathbf{x}_2))) + \mathrm{d}_{\Sigma\mathrm{H}}(\mathrm{p}(P(\mathbf{x}_1)), \mathrm{p}(P(\mathbf{x}_2))), \tag{7}$$

with

$$\mathrm{d}_{\Sigma\mathrm{H}}(A, B) = \sum_{\mathbf{a}\in A} \min_{\mathbf{b}\in B} \|\mathbf{a} - \mathbf{b}\| + \sum_{\mathbf{b}\in B} \min_{\mathbf{a}\in A} \|\mathbf{a} - \mathbf{b}\|. \tag{8}$$

Results. In Fig. 5 the outcome of a single run of the learning-guided restart scheme is plotted against the outcome of 7 standard fitness optimization runs combined (using 24 processor cores in parallel per run, time to complete per run approximately 3 weeks and approximately 30 hours, respectively), and in Fig. 6 the geometry and Mach number distribution of three found solutions per

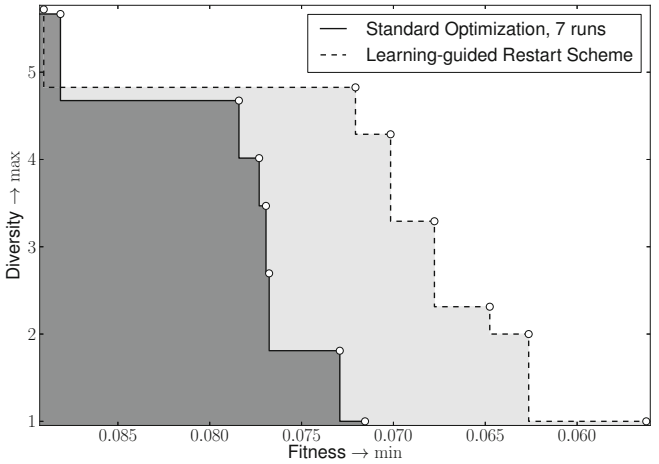
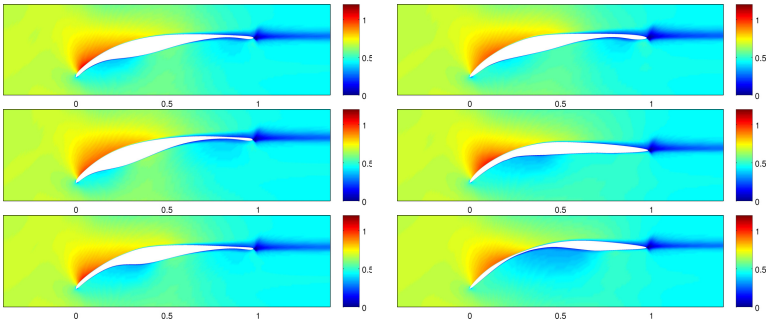


Fig. 5. Fitness vs. Diversity. Per method the quality threshold and diversity of each possible level set in its result set is plotted, governed by the fitness values of the solutions contained in the result set. The learning-guided scheme results in a larger dominated surface in fitness/diversity space, while using only 120k vs. 168k evaluations.



(a) Standard Optimization (b) Learning-guided Restart Scheme

Fig. 6. Geometries Found. The geometries found by standard optimization resemble each other more than those found by the learning-guided scheme. However, all Mach number distributions show high resemblance, seemingly as the left part of the suction side of all blades is similar. Fitness value of solutions in (a) 0.0729, 0.0881, 0.0715, and in (b) 0.0890, 0.0678, 0.0562.

method is displayed. While using less evaluations, the learning-guided restart scheme finds solutions of better quality than the standard optimization runs, and as can be seen in Fig. 6, its solutions are more diverse. However, given that its best solution adheres to the thickness constraints, the learning-guided scheme seems to require adjustment of these as the geometry appears too thin to be feasible in practice.

4 Conclusion and Outlook

An alternating restart scheme was proposed that after convergence of a standard quality optimization phase starts an exploration phase in which is selected on a learning-based interestingness measure [4]. This measure directs the search into areas for which a surrogate model trained on all sampled solutions features high error, but also causes each such area to be left again after a few time steps and thereby sustains exploration.

In earlier work [5] we described a similar alternating scheme that applies a distance-based exploration measure which depends on the similarity to earlier sampled solutions with respect to a domain-specific distance measure. Both schemes were applied to an airfoil optimization task in which the exploration phase continues until the search ends up in infeasible space. A search space containing large pockets of solutions that violate geometric constraints or for which aerodynamic performance cannot be simulated is typical for this problem class.

Compared to multiple standard optimization runs, the learning-guided restart scheme finds solutions with strictly higher quality, and that are more diverse with respect to the domain-specific distance measure mentioned above. For future work a statistical comparison is to be performed with standard optimization, as well as the distance-based scheme. Also, a scheme that generates a random point and then starts moving toward it from a feasible start position is to be considered as a baseline approach to be outperformed, given the assumption that the learning-guided scheme is able to find the boundary of feasible space quicker.

Acknowledgments. The authors thank Giles Endicott for proof reading.

References

1. Cuccu, G., Gomez, F., Glasmachers, T.: Novelty-Based Restarts for Evolution Strategies. In: 2011 IEEE Congress on Evolutionary Computation, CEC 2011, pp. 158–163. IEEE Computer Society (2011)
2. Hansen, N.: The CMA Evolution Strategy: A Comparing Review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a New Evolutionary Computation. STUDFUZZ, vol. 192, pp. 75–102. Springer, Heidelberg (2006)
3. Luciw, M., Graziano, V., Ring, M., Schmidhuber, J.: Artificial Curiosity with Planning for Autonomous Perceptual and Cognitive Development. In: 2011 IEEE International Conference on Development and Learning, ICDL 2011. IEEE Press (2011)
4. Reehuis, E., Olhofer, M., Emmerich, M., Sendhoff, B., Bäck, T.: Novelty and Interestingness Measures for Design-space Exploration. In: 15th Annual Conference on Genetic and Evolutionary Computation, GECCO 2013. ACM Press (2013)
5. Reehuis, E., Olhofer, M., Sendhoff, B., Bäck, T.: Novelty-guided Restarts for Diverse Solutions in Optimization of Airfoils. In: A Bridge between Probability, Set-Oriented Numerics, and Evolutionary Computation, EVOLVE 2013 (2013)

A Trigram Language Model to Predict Part of Speech Tags Using Neural Network

Dinesh Kumar Kashyap[1] and Gurpreet Singh Josan[2]

[1] Department of IT, DAV Institute of Engineering and Technology, Jalandhar, Punjab, India
erdineshk@gmail.com
[2] Department of CS, Punjabi University, Patiala, Punjab, India
josangurpreet@rediffmail.com

Abstract. This paper presents a novel approach of part of speech tagging using neural networks for Punjabi language. To the best of our knowledge neural networks have never been used for the prediction of part of speech tags for Punjabi language. In this paper, a multi layer perceptron neural network tagger with fixed context length has been proposed for tagging of punjabi text. The learning algorithm used for the proposed tagger is error back propagation learning algorithm. The tagged corpus was divided into training and testing data with a randomize function. A feature vector was generated from training data by considering neighboring context for the current word. Trigram model has been used for generating this feature vector for every word in the training data. An overall accuracy of 88.95% is achieved from the tagger. Results shows that the proposed neural network based tagger performs better than existing taggers for punjabi.

Keywords: Part of Speech Tagging, Trigram Model, Punjabi, Multi Layer Perceptron, Neural Network, Tagger.

1 Introduction

Language models were initially developed for speech recognition and machine translation problems. They are used in other natural processing applications like information retrieval, parsing and part of speech tagging. In this paper, we proposed a language model for predicting part of speech tag based on the context of the word, using neural networks. Part of speech tagging is a task of assigning the appropriate part of speech or lexical category to each word in a natural language sentence. It is an initial step in Natural Language Processing (NLP) and is useful for most NLP applications and has a diverse application domain such as Speech Recognition, Speech Synthesis, grammar checker, phrase chunker, machine translation etc. The primary task of tagger is to tag words in the text with proper part of speech tag. But often, part of speech tagging process becomes ambiguous and for a word, multiple tags may exist i.e. the mapping between words to the tags is one-to-many. Part of Speech Tagging can

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 513–520, 2013.

be done using linguistic rules, stochastic models or a combination of both. In the rule based approach, a knowledge base of rule is developed by linguistic to define precisely how and where to assign the various part of speech tags. This approach has already been used to develop the POS tagger for Punjabi language with nearly an accuracy of 80%. During the study on POS taggers for different languages we found that machine learning approaches have been implemented for POS tagging for corpus rich foreign languages, such as Second Order Markov Models based tagger [1], Neural network based tagger [2] for English, [3] for Portuguese, [4] for Chinese, [5]-[6] for Arabic.

In case of Indian Languages particularly Punjabi, we found very limited papers and resources. Punjabi is not a corpus rich language like English. For English and other foreign languages machine learning based approaches have been applied successfully for POS tagging. In case of indian languages specially Punjabi, a rule based tagger [7] and HMM based tagger [8] have been proposed. So, we studied various research papers for foreign and indian languages mainly considering Neural Networks and is presented here. Initially, neural networks were used by [2] for English. Later on other taggers have been successfully proposed for foreign languages like [9][10][11] and [12]. [13] has proposed a neural network based part of speech tagger for Hindi. As per our research, this is the only neural network based tagger available for any indian language. The successful implementation of part of speech tagger using neural networks for foreign and indian languages motivated us to use neural networks for developing a part of speech tagger for punjabi.

The rest of the paper is organized as follows: Section 2 discusses the multi layer perceptron (MLP) network structure, lexicon and input and output representation. Design and implementation of the proposed tagger is discussed in section 3. In section 4, results obtained for punjabi tagger using neural network approach are presented. The paper is concluded in section 5 with future work.

2 Present Approach

The tagging process presented in this paper performs POS tagging at word-level. Although, the word-level approach has a drawback over sentence-level approach, that it is allows unlikely combination of tags but the experiments done by [14] shows that above drawback do not affects the tagging accuracy. In this section, we discussed the approach used for creating a neural network based part of speech tagger for punjabi. It consists of a structure of Multi Layer Perceptron (MLP) network, lexicon and representation of input and output data.

2.1 The MLP Network Structure

The multi layer perceptron network is the type of network which consists of several layers. A typical MLP network must consists of at least three layers, input layer, hidden layer and output layer, where each layer intern consists of basic processing units. The structure of MLP Network designed for the proposed

work is shown in Fig. 2. The input layer of the MLP network consist of words and tags which fall into a fixed length context and target word to be tagged. The output layer of neural network consists of n neurons where each neuron corresponds to one of the tag in the tag set. A tagset proposed by [15] has been used for this work. Error backpropagation learning algorithm is used to assign correct part of speech tag to the target word by adjusting weights of the direct links between the units of the neighboring layers. A hidden layer consisting of $n/2$ neurons are created. In Fig. 1, n_i, n_h and n_o represents the number of nodes in the input, hidden and output layers.

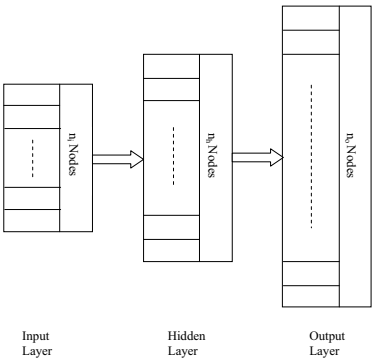


Fig. 1. Structure of MLP Network

The network is trained using error back-propagation algorithm [16] in supervised manner.

2.2 The Lexicon

A lexicon consists of index of the word, the word, occurrences of the word, different tags assigned and probability of occurrence is prepared from training corpus. Here probability is calculated by dividing the number of occurrence of the word by the total no. of words in the corpus. This lexicon is used to generate a feature vector for training and testing of the proposed neural network based tagger. In this work, contextual features are used for creating feature vector with a fixed window size.

2.3 Input and Output Representation

As the input to neural networks is in numeric form so an encoding scheme is devised that converts the textual data from training corpus into numerical data which the neural network can identify and use. Each word from the training corpus is encoded as n-element vector. n is the context word and in our case is 8. The format of input vector is as follows:

$$in = (w_{-2}, tag_{-2}, w_{-1}, tag_{-1}, w_{+1}, tag_{+1}, w_{+2}, tag_{+2}) \quad (1)$$

Here, w_j and tag_j are the indexes of previous and next words along with their tags for the current word w_0. The indexes are find out from words and tags dictionary files generated from training corpus. The input *in* to the network consists of information about the current word w_0. For each word of the training corpus, vector will be generated.

The output *out* is defined as follows:

$$out = (v_1, v_2, ..., v_n) \tag{2}$$

where n is the number of tags and v_j is find out from the training corpus as given in (3)

$$v_j = \begin{cases} 1 & \text{if } tag_j \text{ is the candidate} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

3 Design and Implementation

Figure 2 shows the structure of proposed tagger and the tagging process. Initially, the tagged corpus is divided into training and testing corpus using a randomize function. A training dictionary is generated from training corpus which consists of word index, the word, frequency of its occurrence and tags assigned to the word. This information is maintained in the dictionary for each word of the corpus. Vectorized training file will be generated from the training corpus and training dictionary. As shown in Table 1, trigram contextual features are used for predicting the part of speech tags for unknown words and also for creating feature vectors.

Table 1. Features considered for creating feature vector and predicting POS tags

Word Context	POS Tag Context
w_{-2}	tag_{-2}
w_{-1}	tag_{-1}
w_{+1}	tag_{+1}
w_{+2}	tag_{+2}

Where w_{-2}, tag_{-2} w_{-1}, tag_{-1} are the previous words and their tags and w_{+2}, tag_{+2}, w_{+1}, tag_{+1} are the next words and corresponding tags for the word w_0

A tag remover function was built which removes tags from testing corpus. The purpose to remove tags from testing file it to compare this file after tagging using neural network with the previously available tagged testing file. A comparator function is developed which compares both the testing files and is also used to find out accuracy.

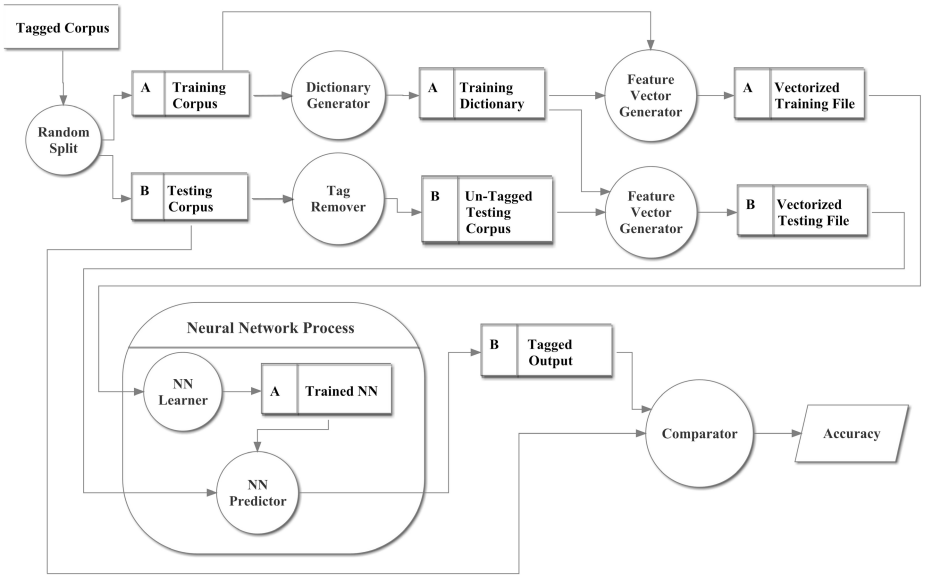


Fig. 2. Structure of proposed tagger and tagging process

3.1 Tagging Process

During the tagging process, we assign a tag to the each word of the sentence. Words may fall under three categories: known words, unknown word and ambiguous words. Known words are the words which are seen in training corpus, unknown words are those which are not seen during training or neural network is not trained for these words and ambiguous words are the words with more than one tag. It is easy to assign a tags for known words. For unknown and ambiguous words, neural network predicts the tag with highest value. As shown in fig. 2, NN learner module is trained using vectorized training file. A trained neural network model will be generated which is passed to NN predictor along with vectorized testing file for the prediction of tag. As discussed in previous section, contextual information is used for training the neural network and for generating neural network based model.

3.2 Implementation

The implementation of multi layer perceptron neural network tagger uses manually tagged punjabi corpus for training and testing. A tagset proposed by [15] consisting of 38 tags was used for the proposed tagger. The neural network is trained using data generated as discussed in the previous section. Error back propagation algorithm [16] is used for training the network. Initially, to calculate the estimated output, forward propagation of the input values is done first. Errors in the form of variation from the expected outputs are calculated and then these errors are circulated backwards in order to adjust the connection strengths.

The gain parameter in the activation function (λ) = 0.8, the learning rate (η) = 0.3 and the output threshold value (ε) = 0.1 during training.

4 Results

A tagged corpus consisting of 26,787 words was used for the proposed work. The text in the corpus is collected from essays, short stories, online sources, news papers etc. The punjabi corpus was tagged with tagset proposed by [15]. This tagging corpus was divided into training data (80.83%) and testing data (19.17%) by using randomize function. Table 2 shows the tag ambiguity analysis obtained from the training corpus. There are lots of occurrences of the words having 4 tags. Table 3 represents the tag wise accuracy achieved by the proposed tagger for some of the tags. The tagger gives accuracy more than 90% for some tags like NN and RB and it gives less than 75% accuracy in case of tags like CC and VBM.

Table 2. Tag Ambiguity Analysis

No. of Tags	Number of Words
1	16415
2	1014
3	123
4	56

Table 3. Tag wise accuracy Analysis

Tag Type	Name of Tags	Accuracy
Noun	NN	99.38%
Adverb	RB	92.12%
Particle	PT	87.00%
Cardinal	QC	83.63%
Adjective	JJ	81.94%
Pronoun	PRP	78.32%
Interjection	INJ	78.00%
Coordinate Conjunct	CC	72.17%
Verb Main	VBM	71.75%

The overall accuracy achieved by the proposed tagger is 88.95%.

4.1 Comparison with Exiting Taggers

The proposed MLP neural network based tagger for Punjabi was compared with the existing Punjabi tagger proposed by [7], a rule based tagger and [8], a HMM based tagger. The results shown in Table 4 shows that our tagger performed better than the already existing taggers for Punjabi in terms of accuracy.

Table 4. Comparison with existing taggers

Corpus Size	Technique Used	Accuracy
26,479	Rule based	80.61%
26,479	HMM	84.37%
26,787	NN (Proposed)	88.95%

5 Conclusion and Future Scope

In this paper a multi layer perceptron neural network based part of speech tagger with fixed context length is presented. Supervised learning approach with error backpropagation algorithm is used for learning purposes. A feature vector is constructed by considering the window size of 2 (trigram) for the current word i.e. the previous two words along with their tags and next two words along with their tags. A feature vector is generated from the dictionary prepared from training data as discussed in section 3. The proposed model gives better results as compared to existing taggers proposed for punjabi. The tagger can handle unknown and ambiguous words very efficiently with higher accuracy. Considering the limited resources available for Indian languages particularly punjabi, the machine learning based approaches like neural networks support vector machines could be a promising solution for part of speech tagging problem for Indian languages.

During the development of the MLP neural network based tagger for punjabi we considered a fixed length context for generating feature vector. Instead of fixed length context, variable length context could be considered with varying number of hidden layers. Results obtained from proposed tagger shows that MLP neural network tagger performs consistently well for certain part of speech tags but not for all. A rule based approach can be integrated for these class of tags.

References

1. Church, K.W.: A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the Second Conference on Applied Natural Language Processing, pp. 136–143. Association for Computational Linguistics (1988)
2. Schmid, H.: Part-of-speech tagging with neural networks. In: Proceedings of the 15th Conference on Computational Linguistics, vol. 1, pp. 172–176. Association for Computational Linguistics (1994)
3. Marques, N.C., Lopes, G.P.: Using neural nets for portuguese part-of-speech tagging. In: Proc. Of the 5th Cognitive Science of Natural Language Processing Conference (1996)
4. Ma, Q., Sun, M., Isahara, H.: A multi-neuro tagger applied in chinese texts. In: Proc. of 1998 Int. Conf. Chinese Info. Processing, pp. 18–20 (1998)
5. Ahmed, R.S., Chandrasekhar, P., Prasad, M.: Application of multilayer perceptron network for tagging parts-of-speech. In: Proc. Language Engineering Conference, pp. 57–63 (2002)

6. Jabar, H.Y., Sembok, T., Tengku, M.: Design and implement an automatic neural tagger based arabic language for nlp applications. Asian Journal of Information Technology 5, 784–789 (2006)
7. Gill, M., Lehal, G., Joshi, S.: Part of speech tagging for grammar checking of punjabi. The Linguistic Journal 4, 6–21 (2009)
8. Sharma, S., Lehal, G.: Using hidden markov model to improve the accuracy of punjabi pos tagger. In: 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 2, pp. 697–701. IEEE (2011)
9. Lu, B.L., Ma, Q., Ichikawa, M., Isahara, H.: Efficient part-of-speech tagging with a min-max modular neural-network model. Applied Intelligence 19, 65–81 (2003)
10. Perez-Ortiz, J.A., Forcada, M.L.: Part-of-speech tagging with recurrent neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2001), vol. 3, pp. 1588–1592. IEEE (2001)
11. Maragoudakis, M., Ganchev, T., Fakotakis, N.D.: Bayesian reinforcement for a probabilistic neural net part-of-speech tagger. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 137–145. Springer, Heidelberg (2004)
12. Marques, N.C., Lopes, G.P., Faculdade, L., Tecnologia, C.: A neural network approach to part-of-speech tagging. In: Proceedings of the Second Workshop on Computational Processing of Written and Spoken Portuguese, pp. 1–9 (1996)
13. Parikh, A.: Part-of-speech tagging using neural network. In: Proceedings of ICON 2009: 7th International Conference on Natural Language Processing (2009)
14. Merialdo, B.: Tagging english text with a probabilistic model. Computational Linguistics 20, 155–171 (1994)
15. Kumar, D., Josan, G.S.: Developing a tagset for machine learning based pos tagging in punjabi. International Journal of Applied Research on Information Technology and Computing 3, 132–143 (2012)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L., PDP Research Group, C. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)

Handling Different Levels of Granularity within Naive Bayes Classifiers

Kemal Ince[1] and Frank Klawonn[2,3]

[1] Volkswagen AG
Komponenten-Werkzeugbau
Gifhornerstr. 180, D-38037 Braunschweig, Germany
[2] Department of Computer Science
Ostfalia University of Applied Sciences
Salzdahlumer Str. 46/48, D-38302 Wolfenbuettel, Germany
[3] Bioinformatics and Statistics
Helmholtz Centre for Infection Research
Inhoffenstr. 7, D-38124 Braunschweig, Germany

Abstract. Data mining techniques usually require a flat data table as input. For categorical attributes, there is often no canonical flat data table, since they can often be considered in different levels of granularity (like continent, country or local region). The choice of the best level of granularity for a data mining task can be very tedious, especially when a larger number of attributes with different levels of granularities is involved. In this paper we propose two approaches to automatically select the granularity levels in the context of a naive Bayes classifier. The two approaches are based on the χ^2 independence test including correction for multiple testing and the minimum description length principle.

1 Introduction

Nowadays it is usual to use data mining techniques to analyze large data sets. A multiplicity of data mining methods with different focuses is in use. The key aspect of these methods depends beside of the analysis intention on the data structure which has to be analyzed. Due to the technological progress nowadays it is very simple to store large data sets in special "data storage systems". Such a data storage system could be for example a data warehouse system. Especially in data warehouse systems where OLAP (online analytical processing) techniques are used it is characteristic to store attributes in the data including different granularity levels. Various methods, for example like drill down or roll up, were developed to navigate in these different granularity levels. Also many more or less efficient approaches to analyze such large databases are known. It is possible to use different data mining methods to get very representable results, which depend on the intention of the analysis case.

But a central problem of almost all established data mining methods is that different granularity levels in the data are usually not taken into account during the analysis step. Most data mining techniques assume a flat data table and not a data cube with different granularity levels for each attribute as they are common in data warehouse systems. If these granularity levels were not involved into the analysis step it has serious consequences in getting acceptable analysis results. In this paper, we propose a

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 521–528, 2013.

technique how a naive Bayes classifier can handle the problem of different granularity levels during the construction of model.

In Section 2 we give a short introduction to the problem area at Volkswagen and also describe conventional approaches to handle – or better ignore – different levels of granularity. The theoretical backround of approach for the naive Bayes classifier is given in Section 3. Section 4 briefly evaluates our methods on theoretical grounds as well as on real data and the final conclusions explain that our methods can also be used as a feature selection technique for naive Bayes classifiers.

2 Motivation

The motivation to develop our approach comes from the components toolshop of Volkswagen AG in Braunschweig. This business area is responsible for producing tools to be used in other divisions of the Volkswagen company like the serial production. Characteristic for the components toolshop is the different product range whose manufacturing process is managed by a special ERP (enterprise resource planning) system based on a relational data base. Analyzing this data by using specific data mining methods assumes the consideration of different granularity levels.

The choice of the right level of granularity for each attribute can be considered as a special case of feature selection. Feature selection includes a big area of research and delivers efficient techniques to choose the attributes to generate the "best" data mining model. Before we introduce our developed approach we want to give a short outlook on feature selection. Feature selection deals with choosing a subset of attributes which – in the ideal case - contains only the relevant and no redundant attributes. Feature selection can be categorized into wrapper and filter methods. Characteristic for filter methods is that the feature selection is done before the data mining algorithm is applied. In contrast to this, the wrapper approach is included into the data mining algorithm. The result of the data mining algorithm with different subsets of attributes is evaluated and the subset of chosen attributes that yields the best result is selected. As mentioned in [1] the wrapper methods do not always deliver the best or efficient way to distinguish between the different granularity levels. Hence the first detected weakness of the existing popular feature selection techniques is that the different granularity levels which could exist were not taken into account during the analysis phase. The introduced approach to use data mining algorithms which contain their own feature selection strategies while constructing the model – like regression trees for example – eliminates this weakness. However this approach would as mentioned in [1] imply the generation of a complex model using the refined granularity levels which is characteristic for an overfitted model. A measure to avoid this bias by generating regression trees was introduced by using the minimum description length (MDL) approach during the splitting step. As mentioned in the conclusions it is necessary to check whether a adaption of the Integrated Approach to Granularity Level Selection as described in the previous publications into other data mining approaches is possible or not. In this paper we introduce the adaption into the naive Bayes classifier.

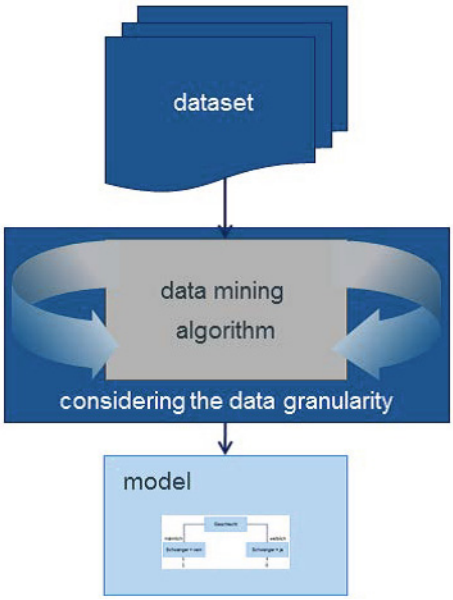


Fig. 1. Integrated Approach to Granularity Level Selection (IAGLS)

3 Theoretical Background and Algorithms

We consider classification problems where we want to predict the value of a categorical attribute C based on the values of other categorical attributes $A_1, \ldots, A_k$. We focus on naive Bayes classifiers that are based on Bayes theorem and calculate the probability or likelihood for each class $c \in \mathrm{dom}(C)$ – $\mathrm{dom}(C)$ denotes the domain i.e. the set of all possible values of attribute C – by

$$P(C = c|A_1 = a_1, \ldots, A_k = a_k) \;=\; \frac{P(C = c) \cdot \prod_{i=1}^{k} P(A_i = a_i|C = c)}{\prod_{i=1}^{k} P(A_i = a_i)}. \quad (1)$$

The classifier is called naive because it assumes that the attributes $A_1, \ldots, A_k$ are assumed to be independent. Although this is not a realistic assumption, naive Bayes classifiers often show quite good or at least reasonable performance [2].

The tuple $(a_1, \ldots, a_k)$ is usually assigned to the class c for which Eq. (1) yields the highest probability [3]. This means that all misclassifications are considered to be equally bad. Of course, one can also introduce a cost function that assigns individual costs to different misclassification, e.g. it might be worse to wrongly classify an object from class c_1 to class c_2 than to wrongly classify an object from class c_2 to class c_1 which would expressed by lower misclassification costs for the latter case (see for instance [4]).

It should be noted that the denominator in Eq. (1) is only a normalisation factor that guarantees that the sum of the probabilities over all possible values of the class attribute

C yields 1. This denominator does not depend on c and can therefore be omitted for the classification decision. We will also do this and only consider the likelihood

$$P(C = c|A_1 = a_1, \ldots, A_k = a_k) \;=\; P(C = c) \cdot \prod_{i=1}^{k} P(A_i = a_i|C = c) \qquad (2)$$

instead of Eq. (1).

It should be mentioned that continuous attributes can also be incorporated into the naive Bayes classifier easily [5]. But since we are especially interested in attributes with different levels of granularity, we do not take continuous attributes into account, since they can have only one level of granularity. When continuous attributes are also present, they can be incorporated into our approach in the usual way.

We exploit the simplicity of the naive Bayes classifier that attributes used for the prediction are considered to be independent. Therefore, we determine their level of granularity also independently. Let us consider a single attribute A for which we have different levels of granularity. The coarsest level would be to ignore attribute A completely for the prediction of attribute C completely. This would mean that attribute A always assumes a constant value a with probability 1. At the first non-trivial level of granularity, attribute A can assume the values $a_1, \ldots, a_{\ell_1}$. This level might be refined to $a_{1,1}, \ldots, a_{1,\ell_{2,1}}, \ldots, a_{\ell_1,1}, \ldots, a_{\ell_1,\ell_{2,\ell_1}}$, i.e. the attribute value a_1 can be refined to the values $a_{1,1}, \ldots, a_{1,\ell_{2,1}}$ etc. This refinement corresponds to the drill-down operation in the OLAP setting [6].

Two approaches to finding the right level of granularity will be discussed. One is based on the χ^2 test, the other one on the minimum description length principle. We will use the χ^2 test to compare two levels of granularity in order to decide whether the coarser or the finer level should be chosen. The approach based on the minimum description length principle measures the suitability of each level of granularity separately and the different levels can be compared on the basis of these values.

3.1 An Approach Based on the χ^2 Test

Let us consider two levels of granularity of the same attribute A. On the coarser level, A can assume the values $a_1, \ldots, a_\ell$, on the refined level A can assume the values $a_{1,1}, \ldots, a_{1,\ell_1}, \ldots, a_{\ell,1}, \ldots, a_{\ell,\ell_\ell}$, i.e. the attribute value a_1 can be refined to the values $a_{1,1}, \ldots, a_{1,\ell_1}$ etc.

Table 1. Contingency table obtained from the refinement of attribute value a_1

	dom(C)		
a_1 refined	c_1	...	c_s
$a_{1,1}$	$n_{1,1}$	...	$n_{1,s}$
⋮	⋮	⋮	⋮
a_{1,ℓ_1}	$n_{\ell_1,1}$	...	$n_{\ell_1,s}$

We are interested in the question whether the refined level yields a sufficiently higher correlation than the coarser level. Considering the first value a_1 of the coarser level, we obtain a contingency table as shown in Table 1. The entries in the table indicate the number of instances in the training data set with the corresponding combination of values, i.e. $n_{1,1}$ is the number of instances with value $a_{1,1}$ for attribute A and c_1 for the target attribute C.

In the worst case, in terms of the naive Bayes classifier, the refined values for attribute value a_1 are independent of the class values of the target attribute C. This would mean that choosing the refined level would – at least for the attribute value a_1 – not bring any advantages for the classification. Therefore, we apply a χ^2 independence test and take the resulting p-value into account. Of course, we must consider ℓ contingency tables, one for each value of the attribute A on the coarser level. Therefore, we obtain ℓ p-values. There are two ways to decide whether the attribute A should be refined.

(a) A very conservative approach would require that the refinement for all values on the coarser level leads to a significant improvement, i.e. we would expect dependencies with class attribute C in each of the ℓ contingency tables. This means that we would consider the worst of the ℓ p-values and refine the attribute A only if the largest p-value is still significant, say the p-value must be below 5%.

(b) It is sufficient that we find in at least one of the contingency tables a strong dependency between the class attribute and the refined values. In this case, it is necessary to take the problem of multiple testing [7] into account, since we apply the χ^2 test ℓ times. The p-values must be corrected in order to avoid random correlations between the refined attribute and the class attribute. Techniques like Bonferroni (see for instance [7]), Bonferroni-Holm [8] or false discovery rate (FDR) correction [9] should be applied. For reasons of simplicity, we use FDR correction here and multiply the p-values with the number of contingency tables and use the refined attribute values only if at least one of the corrected p-values[1] is significant enough, say below 5%.

3.2 An Approach Based on the Minimum Description Length Principle

The minimum description length principle (MDL) (see for instance [10]) is a model selection technique to overcome the problems of overfitting. The underlying idea of MDL is the interpretation of a model – in our case the naive Bayes classifier – as a kind of coding for the data. If the naive Bayes classifier perfectly predict the correct classes without mistakes, it would be sufficient to know the naive Bayes classifier and the values for the attributes A of each record to determine the class, i.e. its corresponding value for the attribute C. If the naive Bayes classifier predicts the classes for most of the records correctly then it is sufficient to know the naive Bayes classifier and the classes for those records for which the naive Bayes classifier makes a wrong prediction. MDL prefers the model that provides the shortest coding.

In our case of the naive Bayes classifier, we might choose very refined levels for all attributes and obtain a Bayes classifier that makes no or only a few wrong predictions.

[1] These corrected p-values are also called q-values in statistics.

In this case, the coding for the corrections would be small. However, since the attributes are considered on a refined level, the coding to store this larger number of probability values would make the whole coding inefficient.

The other extreme would be a naive Bayes classifier using attributes only a very coarse or the coarsest level. Then the coding for the few required probabilities would be small. However, the coding for the required corrections of the wrong predictions would be enormous.

When we consider a single attribute A with possible values $a_1, \ldots, a_\ell$. If the naive Bayes classifier uses only this attribute, we need to store the $\ell \cdot s$ probabilities $P(a_i|c_j)$ if we have s classes $c_1, \ldots, c_s$. We would also have to store the prior probabilities for the classes. But we can ignore them for the considerations in the context of MDL because any naive Bayes classifier will have to store them, no matter whether it is based on a coarse or a refined level. In addition the probabilities $P(a_i|c_j)$, it is also necessary to store the correct class for those records for which the naive Bayes classifier makes a wrong prediction.

Let us first consider how many bits are needed for the correction of the wrongly predicted records. Since we have s classes, we need $\lceil \log_2(s) \rceil$ to code the s classes. If n_{wrong} denotes the number of wrongly classified records, we need altogether $\lceil \log_2(s) \rceil \cdot n_{\text{wrong}}$ bits to the necessary corrections.

The storage of the probabilities depends on the required precision. Here we assume that the probabilities should be stored with a precision of prec binary digits. In this paper we use prec $= 5$. A refined level has more values, therefore the probabilities $P(a_i|c_j)$ tend to be smaller and the refinement obtains a penalty in this way although it might lead to (slightly) better predictions. The number of bits to store the naive Bayes classifier including the corrections is therefore

$$\lceil \log_2(s) \rceil \cdot n_{\text{wrong}} + \lceil -\log_2(\text{minprob}) + \text{prec} \rceil \cdot s \cdot \ell. \tag{3}$$

For each attribute of the naive Bayes classifier we choose the level of granularity for which Eq. (3) yields the lowest value.

4 Evaluation

We have tested our approaches on random data sets where we have generated attributes with a high correlation to the classes. We have then refined these attributes randomly and also coarsened these attributes. In all cases, both approaches based on the χ^2 independence test and based on MDL could identify the correct level of granularity. For reasons of confidentiality we cannot discuss the results on the real world data set from the components toolshop.

In order to better illustrate the differences between the two approaches, we consider a very simple theoretical setting which is more in the spirit of attribute selection than of granularity level selection. But this restriction makes it easier to visualise the results. Consider an attribute with a trivial level of granularity where the attribute always assumes a constant value. This actually means that this attribute does not provide any information to the classification problem in this coarse level of granularity. In other words, if we

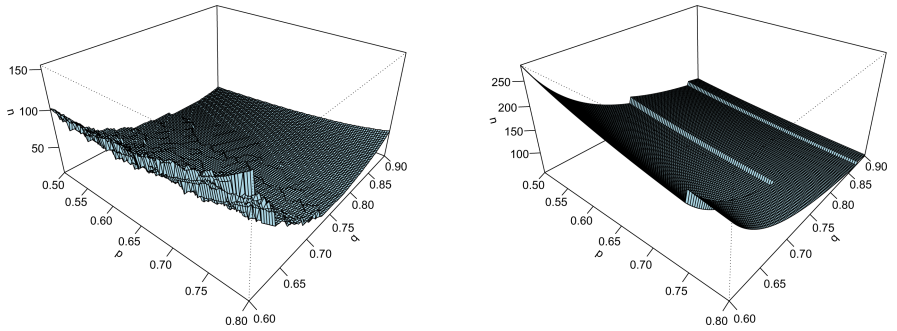


Fig. 2. The required number n of records to favour the refined attribute given a binary refinement split with proportions p and $(1-p)$, where a proportion of q records of one class is assigned to the first split. The prior probabilities for this two-class problem are assumed to be equal. The left graph is based on the χ^2 independence test, the right one on the MDL approach.

choose this coarse level of granularity, the attribute is not selected. We assume that we have a two-class problem with equal prior probabilities for the two classes.

The attribute can be "refined" to two values. The data set is split by these two values into one part containing a proportion of $p > 0.5$ and another part containing a proportion of $(1-p)$ of the records. The first part is assumed to contain a proportion of $q > 0.5$ of the first class. Figure 2 shows how many records n the data set must contain in order to favour the split depending on the values of p and q. The left graph represents the approach based on the χ^2 independence test, the right one for the MDL approach. The χ^2 independence test approach needs a smaller sample size to favour the refined level. It is also more sensitive to a variation of p and q.

After these theoretical considerations, we evaluate our approach with data from the ERP-system of the components toolshop containing 40,449 instances. The target attribute is the manufacturing time for a product. The manufacturing time is given in eight discrete time intervals, so that it can be considered as a categorical attribute. The relative frequency for these eight different time intervals ranges from 3.5% up to 49.7%. For reasons of simplicity, we discuss only one attribute, the machine group, which is given in two different levels of granularity. The coarse level has only three possible values, the refined one 95. This extreme refinement might easily lead to overfitting. However, both approaches, the χ^2 test (p-value less than 10^{-15}) and also the MDL approach (56,682 for the coarse and 40,232 for the refined attribute) clearly favour the refined attribute.

We then constructed a naive Bayes classifier with the coarse machine type attribute and with the refined one. In both cases, other attributes were also used for the prediction of the manufacturing time. We applied 10-fold cross-validation in both cases. With the coarse attribute, a misclassification rate of 31.9% was achieved, whereas the use of the refined attribute did not lead to overfitting, but yielded a misclassification rate of 14.1%. So the recommendation of our χ^2 test and MDL approach to use the refined attribute was correct here.

It should be noted that other attributes with different levels of granularity are also involved in this classification problem. A detailed discussion of all these attributes would be out of the scope of this paper. But we can observe similar effects as for the machine type, i.e. the recommended level of granularity leads to the best performance of the naive Bayes classifier.

5 Conclusions

We have proposed two approaches to automatically choose the level of granularity of categorical attributes for a naive Bayesian classifier. These approaches can also be used for attribute selection for a naive Bayes classifier as shown in the contexts of Figure 2.

References

1. Ince, K., Klawonn, F.: Decision and regression trees in the context of attributes with different granularity levels. In: Borgelt, C., Gil, M.Á., Sousa, J.M.C., Verleysen, M. (eds.) Towards Advanced Data Analysis. STUDFUZZ, vol. 285, pp. 331–342. Springer, Heidelberg (2012)
2. Hand, D., Yu, K.: Idiot's bayes—Not so stupid after all? International Statistical Review 69, 385–398 (2001)
3. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. 29, 103–130 (1997)
4. Berthold, M., Borgelt, C., Höppner, F., Klawonn, F.: Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data. Springer, London (2010)
5. John, G., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, pp. 338–345. Morgan Kaufmann (1995)
6. Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Rec. 26, 65–74 (1997)
7. Shaffer, J.P.: Multiple hypothesis testing. Ann. Rev. Psych. 46, 561–584 (1995)
8. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70 (1979)
9. Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society. Series B (Methodological) 57, 289–300 (1995)
10. Gruenwald, P.D.: The Minimum Description Length Principle. The MIT Press, Cambridge (2007)

Genetic Algorithm on GPU Performance Optimization Issues

Andrius Paukštė

Faculty of Mathematics and Informatics
Vilnius University
Vilnius, Lithuania
andrius.paukste@mif.vu.lt

Abstract. The aim of this paper is to investigate genetic algorithm execution on graphics processing unit performance issues and to develop techniques on how to speed up execution by optimizing algorithm execution path and data allocation. The paper presents methods to improve genetic algorithm performance by achieving higher hardware utilization and efficient task distribution between a graphics processing unit and a central processing unit.

Keywords: genetic algorithms, graphics processing unit, parallel computing.

1 Introduction

1.1 Genetic Algorithms

Genetic algorithms (GAs) are machine learning techniques inspired by natural evolution processes [1]. A population in GA is a set of individuals. Each individual represents one of the possible solutions of the given problem. The goal of GA is to find the optimal solution or the best individual of the population. At each iteration, GA performs genetic operations on the current population (selection, crossover, and mutation) to produce a new population.

GAs proved their effectiveness in many scientific or applied applications. However, in cases that involve a high number of parameters and/or a complex fitness function evaluation, GAs become both time and computation intensive. The parallelization of GA is a natural way to reduce computing time and improve algorithm performance.

1.2 Graphics Processing Unit

Over the past few years, general purpose computations on graphics processing units (GPUs) have become increasingly popular. GPU architecture makes them extremely efficient for compute-intensive and highly parallel applications.

Modern GPUs have more than a thousand simple computing cores on a single chip. However, architecture makes them most efficient for single instruction, multiple data

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 529–536, 2013.

(SIMD) [10] operations. It makes the path of an algorithm and data allocation extremely important for achieving high performance.

Modern GPUs become more complex and introduce parallel kernel (GPU program) execution and asynchronous memory copy operations to and from GPU device. We will employ those new parallel computing features in this paper to develop more efficient parallel GA than before.

1.3 Genetic Algorithms on GPU

Previous research of [2] and [4] show that GPUs can be effectively leveraged for evolutionary computations. [5] and [6] propose execution of the whole GA on GPU. However, [6] concludes that the implementation of the selection procedure is not effective on GPU in case we need to collect statistical information on the whole population. To solve this problem, [2] uses an island-based genetic algorithm where all the individuals are divided between subpopulations (islands) with limited migration between them. Every separate island is executed on different block of the GPU threads, and expensive interblock communication is limited.

However, there is a lack of research discussing heterogeneous GA execution on GPU and CPU in parallel. A heterogeneous algorithm described in section 2 will utilize CPU computing power in parallel with GPU and perform genetic procedures such as selection, crossover, mutation, and replacement on the CPU side. Splitting operations between CPU and GPU will allow us to fully utilize computational power of every device.

Finally, especially with a big population size, representation of data in GPU memory has a significant impact on branching in code. In section 3, we will analyze the effect of presorting of chromosome vectors before the fitness function evaluation on GPU.

2 GA Performance Optimization Issues

Below we will introduce a heterogeneous GA with dividing tasks between CPU and GPU devices. We propose methods to execute GA in parallel on both CPU and GPU devices. We assume that the fitness function evaluation is the most computing-intensive task in algorithm and requires more time to perform on a GPU device than to perform genetic operations on CPU.

2.1 Optimization of Parallel GA Execution on GPU and CPU

To utilize the full computational power of GPU and CPU devices, we divide tasks that are better appropriate for each device.

Fitness Function Evaluation. The complex fitness function evaluation is often more suited to be performed on GPU, especially if it is possible to perform a single-parameter vector evaluation on more than one GPU thread in parallel.

Mutation Function. The mutation function may be performed well on both CPU and GPU devices because of the simple parallelization. However, in case the fitness function requires relatively more computational power than genetic operations, there is an advantage in executing the mutation function on already evaluated chromosomes on CPU in parallel with the fitness function evaluation on GPU.

Selection and Replacement Function. The selection and/or replacement function is more suited for CPU because it is complicated to fully utilize a GPU device. [4] proposes a parallel multiobjective evolutionary algorithm (MOEA) on GPU. The proposed algorithm is almost fully executed on GPU, except for the nondominated selection. The authors conclude that this part of the algorithm is not suitable for parallelization on GPU because of the difficulty to achieve high hardware utilization.

Below we analyze selection procedures [3] that will allow executing them in parallel with fitness evaluation. Only selection methods that do not require knowing the finesses of all the individuals will allow executing them in parallel. After, we propose parallel execution procedures for a few of them.

The following are selection methods that allow the parallel execution with the fitness function evaluation:

Tournament selection (binary, larger, or correlative tournament)

We propose the following tournament selection execution in parallel method:

1. We randomly choose N individuals from the population.
2. If the fitness values of all the individuals are known at the moment of tournament, we perform tournament selection between the individuals. In case there is at least one individual with an unknown fitness value, we wait until all the fitness values are known.

Selection generation

The offspring of the parent population becomes the next population.

The following are selection methods that don't allow parallel execution with the fitness function because of the requirement of knowing all the fitness values (to compute ranks or sum) before the start of the selection procedure:

Fitness proportionate selection (roulette wheel selection)
Stochastic universal sampling
Deterministic sampling
Stochastic remainder sampling
Range selection

2.2 Population Construction Method

In this section we propose population construction method based on partially known fitness function evaluation results.

We suppose that for a large number of individuals and a complex fitness function evaluation on GPU, we are able to transfer part of the evaluation results before the end of the evaluation.

We suppose S_p is the size of population P, and S_{ev} is the size of the part of population with the already evaluated fitness function on GPU and transferred results to the

host. Depending on the selection and replacement procedures, when $S_{ev} > S_p/2$, we are able to construct part of a new population of size $S_n = S_{ev} - S_p/2$ with. Table 1 show the example of tournament selection method when fitness of the 75% of chromosomes is known.

Table 1. Example of tournament selection when fitness of 75% of the chromosomes is known

Nr. of the chromosome	Is fitness known?	Random pairs	Population after tournament	Crossover random pairs	Population after crossover
1	Yes	8/2	?	4/7	1
2	Yes	4/1	1	1/5	2
3	Yes	7/4	?	8/3	?
4	Yes	6/5	5	6/2	?
5	Yes	3/2	2		?
6	Yes	5/8	?		?
7	No	7/6	?		?
8	No	1/3	1		?

2.3 Parallel Heterogeneous CPU/GPU GA Execution Algorithm

First, we create initial population and start evaluation on GPU. After transferring evaluation results of size S_{ev}, we will perform selection, mutation, crossover, and replacement procedures on the CPU side and send part of the new population of size S_n back to the GPU. After completing the evaluation procedure of the previous population, GPU will start to evaluate a part of the new population S_n, and the evaluation work on GPU won't stop. After the results of the previous population will be transferred to the host the new population will be fully constructed on the CPU side and ready for evaluation.

The following is the proposed heterogeneous CPU/GPU GA:

1. Initialize initial population on GPU.
2. Evaluate the S_{ev} part of population on GPU.
3. Evaluate the $S_p - S_{ev}$ part of population on GPU and in parallel on CPU:
   (a) Transfer the data evaluated in the previous step from GPU device to host.
   (b) Apply selection procedure to select the new population of size S_n.
   (c) Apply mutation, crossover, and replacement procedures.
   (d) Transfer part of the new population of size S_n to GPU device.
4. Evaluate part of the new population of size S_n on GPU. In parallel on CPU:
   (a) Transfer evaluated part of previous population to from GPU device to host.
   (b) Apply selection procedure to select the last part of the new population of size $S_p - S_n$
   (c) Apply mutation, crossover, and replacement procedures.
   (d) Transfer the last part of new population of size $S_p - S_n$ to GPU device.
5. Until terminating condition is reached return to step 2.

Figure 1 represents the scheme of the proposed algorithm.

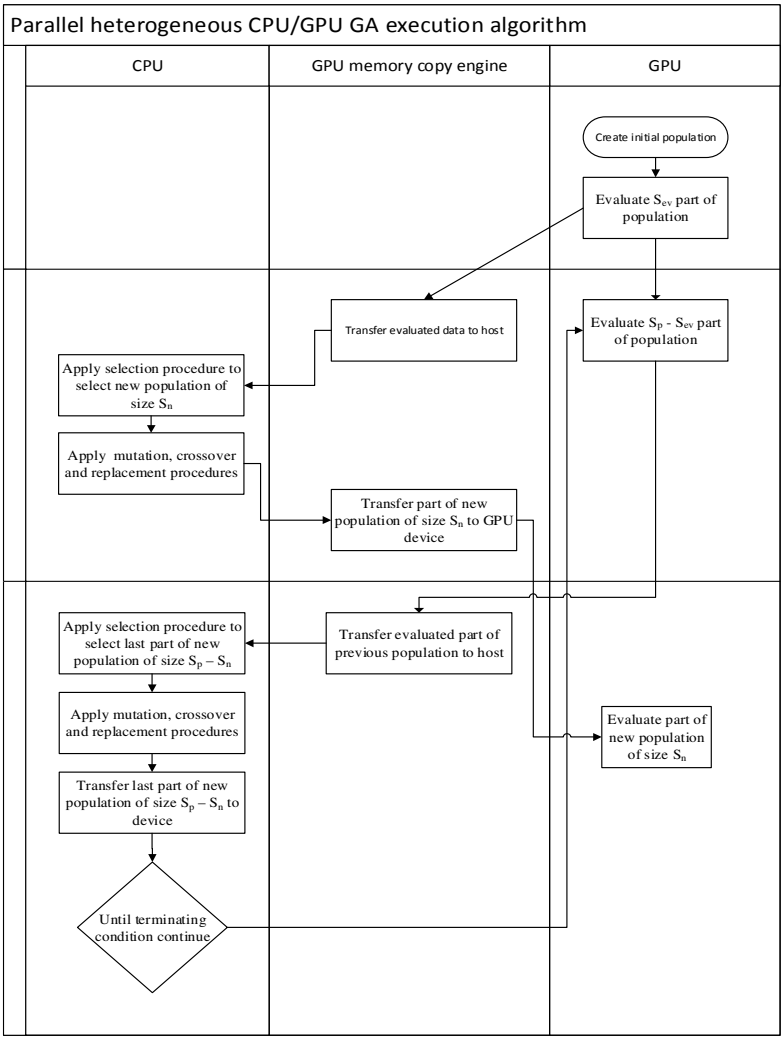


Fig. 1. Parallel heterogeneous CPU/GPU GA execution algorithm

3 Divergence Optimization

The week point of SIMD architecture is a divergence problem when processing units follow different path after a branch instruction.

In the previous work [7], we have shown that Monte Carlo optimization parameters presorting before the execution on GPU may significantly (more than three times) reduce computing time. Sorting parameters vectors (chromosome values in case of GA) before the execution will reduce branches in GPU warps and allow neighboring

threads to execute more similar code than before presorting. Table 2 shows the example of population sorting. Each line represents one chromosome.

Table 2. Population fragments before and after sorting.

Population before sorting:	Population after sorting:
139 520 359 197	94 150 395 430
696 348 736 822	94 520 359 197
94 150 395 430	112 243 331 809
525 586 683 888	112 755 20 397
530 262 704 391	112 916 683 888
861 44 442 769	114 262 704 391
…	…

We propose to apply the parameters sorting technique after selection (steps 3c and 4c) procedure on the CPU side.

4 Experiments

4.1 Experiment Description

Although the optimization method based on GA may be beneficial in many areas, in this paper, we analyze the application for modeling financial risks [8]. The fitness function *f* represents a systematic trading portfolio:

$$r1, r2, \ldots, rM = f(p1, p2, \ldots, pN),$$

where $p1, p2, \ldots, pN$ are the portfolio allocation parameters and $r1, r2, \ldots, rM$ are the risk measures. Following risk measures are estimated:

1. VaR (Value at Risk);
2. Sharpe ratio;
3. PnL (Profit & Loss);

4.2 Simulation Settings

Population size: 256 000 Number of the fitness function parameters: 9

Number of the possible unique fitness function parameters vectors: 100^9

GA evolution steps: 100

GPU: NVIDIA Tesla M2090 CPU: Intel Xeon E5649 6-Core at 2,53GHz

4.3 Experiments Results

In Table 3, we show the real-time simulation results on GPU and CPU with different optimization techniques. We measure actual GA execution time including initial population generation, data transfers between CPU and GPU, etc.

Control flow execution efficiency represents the amount of active threads per warp for each instruction divided by the total number of threads per warp. The measure does not take into account a duration of instructions, however together with execution time shows significant improvement in branching efficiency.

Table 3. Simulation results

	Kernel 1, speedup	Kernel 1, control flow execution efficiency	Kernel 2, speedup	Kernel 2, control flow execution efficiency
Execution on CPU (6 cores, OpenMP)	1		1	
Execution on GPU	3,47	0,23	6,82	0,41
Heterogeneous CPU/GPU algorithm	3,88	0,23	7,24	0,40
Execution on GPU with presorting	7,7	0,41	10,3	0,76
Heterogeneous CPU/GPU algorithm with presorting	8,1	0,39	11,62	0,71

5 Conclusions

In this paper, we discuss GA execution on GPU acceleration issues. We propose two methods to achieve higher GA performance on graphic devices.

First, a heterogeneous GA execution algorithm has been presented. Algorithm allows increasing GPU computing efficiency by separating tasks between CPU and GPU.

Second, we have presented the population presorting algorithm, which allows increasing branching efficiency in code executed on GPU.

Finally, both presented methods show significant impact on algorithm performance (up to two times) and are applicable in evolutionary computations on GPU in many areas.

Acknowledgment. The author acknowledges support by project "Promotion of Student Scientific Activities" (VP1-3.1-ŠMM-01-V-02-003) from the Research Council of Lithuania. This project is funded by the Republic of Lithuania and European Social Fund under the 2007-2013 Human Resources Development Operational Programme's priority 3.

The reported study was supported by the Supercomputing Center of Lomonosov Moscow State University [9]. The author would also like to express his appreciation to the Laboratory of Parallel Information Technologies lead by Prof. Vladimir Voevodin.

References

1. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley (1989)
2. Pospichal, P., Jaros, J., Schwarz, J.: Parallel Genetic Algorithm on the CUDA Architecture. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 442–451. Springer, Heidelberg (2010)
3. Sivaraj, R., Ravichandran, T.: A review of selection methods in genetic algorithm. Int. J. Eng. Sci. Technol. 3(5), 3792–3797 (2011)
4. Wong, M.L.: Parallel multi-objective evolutionary algorithms on graphics processing units. In: GECCO 2009: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, pp. 2515–2522. ACM, New York (2009)
5. Yu, Q., Chen, C., Pan, Z.: Parallel genetic algorithms on programmable graphics hardware. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 1051–1059. Springer, Heidelberg (2005)
6. Wong, M.L., Wong, T.T.: Implementation of Parallel Genetic Algorithms on Graphics Processing Units. In: Gen, M., Green, D., Katai, O., McKay, B., Namatame, A., Sarker, R.A., Zhang, B.-T. (eds.) Intelligent and Evolutionary Systems. SCI, vol. 187, pp. 197–216. Springer, Heidelberg (2009)
7. Paukste, A.: Monte Carlo optimisation auto-tuning on a multi-GPU cluster. In: 2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 894–898. IEEE (2012)
8. Brabazon, A., O'Neill, M.: Biologically inspired algorithms for ¯nancial modelling. Natural Computing Series. Springer, Berlin (2006)
9. Sadovnichy, V., Tikhonravov, A., Voevodin, V.I., Opanasenko, V.: "Lomonosov": Supercomputing at Moscow State University. In: Contemporary High Performance Computing: From Petascale toward Exascale (Chapman & Hall/CRC Computational Science), pp. 283–307. CRC Press, Boca Raton (2013)
10. Flynn, M.J.: Some computer organizations and their effectiveness. IEEE Transactions on Computers C-21, 948–960 (1972)

Mutual Information for Performance Assessment of Multi Objective Optimisers: Preliminary Results

Noura Al Moubayed, Andrei Petrovski, and John McCall

School of Computing
The Robert Gordon University
Aberdeen, UK
n.al-moubayed@rgu.ac.uk

Abstract. Solving multi-objective problems usually results in a set of Perto-optimal solutions, or a Pareto front. Assessing the quality of these solutions, however, and comparing the performance of different multi-objective optimisers is still not very well understood. Current trends either model the outcome of the optimiser as a probability density function in the objective space, or defines an indicator that quantify the overall performance of the optimiser. Here an approach based on the concept of mutual information is proposed. The approach models the probability density function of the optimisers' output and use that to define an indicator, namely the amount of shared information among the compared Pareto fronts. The strength of the new approach is not only in better assessment of performance but also the interpretability of the results it provides.

1 Introduction

Many real-world applications often involve optimisation of multiple, competing objectives in large search spaces [1]. It is therefore an important task to effectively and simultaneously address multiple optimisation objectives by identifying a set of well-distributed Pareto optimal solutions that yield good values for each objective. Population-based metaheuristics (e.g. Genetic Algorithms) have been developed to facilitate an efficient search in multi-dimensional solution spaces, the feasible regions within which are determined by a set of (often non-linear) constraints. However, instead of obtaining infinite number of Pareto optimal solutions, which is a time consuming and resource demanding task, it is often preferable to search for a set of representative solutions that closely approximate the true Pareto front being uniformly distributed along its length [2,3].

As with single objective optimisation, two factors are important when assessing a multiobjective optimiser: the quality of the found solutions, and the time spent to find them. However the stochastic nature of evolutionary algorithms results in the relation between time and quality not fixed, but rather represented by a probability distribution function. In addition having a set of

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 537–544, 2013.

solutions (Pareto front) instead of a single outcome of the multiobjective optimisation process makes quantifying the quality of these solutions much harder. This is added to having multiple runs and the necessity to statistically quantify the behaviour of the optimiser over these runs increases the difficult of quality assessment [5].

This paper introduces a novel approach for performance assessment of mutliobjective optimizers. First a simple mutual-information based method is presented and discussed. A more robust approach is then proposed which looks at PFs as images. To validate and test this method, five problems are tested using three popular multi-objective optimizers and the results are tested using three popular indicators.

2 Background and Related Work

2.1 Multi-objective Optimisation Problems

Solving a multi-objective optimisation problem is challenging because an improvement in one objective often happens at the expense of deterioration in other objective(s). The optimisation challenge therefore is to find the entire set of trade-off solutions that satisfy all conflicting objectives.

Let $F(X) \in \Delta \subset R^m$ be a vector of objectives: $F(X) = \{f_1(X), \ldots, f_m(X)\}$, where $X = \{x_1, x_2, \ldots, x_n\} \in \Omega \subset R^n$ is the vector of decision variables, n is the dimension of solution space, and $m \geq 2$ is the number of objectives. The search space (also called the solution space) refers to the space of decision variables, whereas the objective space is the space where the objective vectors lie. When minimizing $F(X)$, for example, a domination relationship is defined between the solutions as follows: let $X, Y \in \Omega$, $X \prec Y$ if and only if $f_i(X) \leq f_i(Y)$ for all $i = \{1, 2, ..., m\}$, and there is at least one j for which $f_j(X) < f_j(Y)$. X^* is a Pareto optimal solution if there is no other solution $S \in \Omega$ such that $S \prec X^*$. Therefore the Pareto optimality of a solution guarantees that any enhancement of one objective would results in the worsening of at least one other objective. The Pareto optimal set is the set of all non-dominated solutions. The image of the Pareto optimal set in the objective space (i.e. $F(X^*)$) is called the Pareto Front (PF).

2.2 Measuring Quality of Multi-objective Optimizers

Multiobjective evolutionary algorithms are stochastic in nature, due to the random element in the algorithms, i.e. running the algorithm twice would most likely produce a different set of results. For this reason the optimizer should be run several times and the probability density function is then empirically estimated. Comparing two optimizers would then mean comparing their probability density functions which then implicate the issue of statistical hypothesis testing [6].

In the literature, there are two main approaches to assess the quality of produced PFs. The most common one is the indicator approach where a PF is

mapped, using a defined function, to a real number then a standard statistical hypothesis test is applied on the indicator values. The second approach is usually referred to as the attainment function method in which for each objective vector there is a probability p that the produced approximation set contains an objective vector that dominates z. The attainment function then gives a probability estimate of z to be attained in one optimization run with a statistical test procedure to count for all the runs [7].

A comprehensive survey on quality indicators can be found in [5]. However, here we list three recommended indicators in [5] for their theoretical advantage. The indicators are based on different aspects of the data and therefore using them all will provide more ,hopefully complementary, information than using just one.

The inverted generational distance The inverted generational distance, I_{IGD}, [8] measures the uniformity of distribution of the obtained solutions in terms of dispersion and extension. The average distance is calculated for each point of the actual PF (PF_{True}), denoted as A, and the nearest point of the approximated PF (PF_{approx}), denoted as B: $I_{IGD(A,B)} = \frac{(\sum_{a\in A}(\min_{b\in B}\|F(a)-F(b)\|^2))^{1/2}}{|A|}$ The indicator usually gives a fair evaluation of the produced PF, however it can be prune to outliers due to the use of the distance measure.

Hypervolume The hypervolume indicator, I_{hv}, [9] measures the volume of the objective space that is dominated by a PF approximation (A). I_{hv} uses a reference point v^* which denotes an upper bound over all objectives. v^* is defined as the worst objective values found in A (i.e. v^* is dominated by all solutions in A). Using the Lebesgue measure (Λ), I_{hv} is defined as: $I_{hv}(A) = \Lambda\Big(\bigcup_{a\in A}\{x\,|a \prec x \prec v^*|\}\Big)$. The advantage of hypervolume is that it is conceptually intuitive, but it can be computationally costly and it requires a reference point which may affect the ordering of pairs in incomparable sets.

ϵ Indicator The ϵ indicator, I_ϵ, [10] measures the minimum distance which a PF approximation (A) has to be translated in the objective space to dominate the actual PF B. The ϵ-Indicator is defined as:$I_\epsilon(A,B) = \min_{\epsilon\in\mathbb{R}}\{\forall b \in B, \exists b'_i - \epsilon \leq b_i, \forall 1 \leq i \leq n\}$. This is a fast indicator to compute and has an intuitive meaning: how much do I need to translate/scale the set A so that it covers the reference set?. However, the choice of reference set can dramatically affect the results.

Using quality indicators is an attractive approach of quality assessment due to its simplicity. It has , however, some shortcomings: 1) each indicator looks at the performance from only one perspective, e.g. spread, diversity, or dominance, which may skew the conclusions drawn. 2) In the case of incomparable PFs an indicator will actually give an inaccurate result. 3) For indicators that use distance functions, outliers can cause a real problem in disturbing the calculation of the indicator. 4) Quality indicators do not take the statistics of the data in the objective space into account.

The approach we take in this paper models the output of the optimizer directly as an empirical probability density function and then calculates the mutual information between the approximated PF and the theoretical one.

3 Methods

3.1 Mutual Information

Intuitively speaking, mutual information measures how much information are shared between two random variables X, Y. In other words how much knowledge of one variable reduces the uncertainly about the other. Formally, mutual information is defined as follows: $I(X,Y) = \sum_Y \sum_X p(x,y) log \frac{p(x,y)}{p(x)p(y)}$, where $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively with summation replaced by an integral in the case of continuous random variables. If the two variables are dependent then $I(X,Y)$ measures the shared information between the two variables and it would be positive. If on the other hand, they are independent then $I(X,Y) = 0$.

To complete the calculation of MI, the joint and marginal probabilities should be calculated. To estimate the joint probability distribution/density function a two dimensional histogram is used [11].

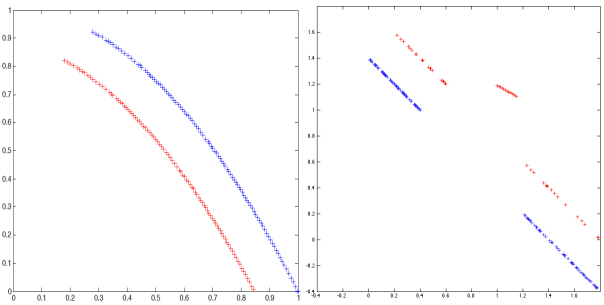
The mutual information as defined so far is not normalized, i.e. it can take any positive value. Here we use a normalized version as defined in [12]:

$$U(X,Y) = 2\frac{I(X,Y)}{H(X) + H(Y)} \tag{1}$$

where $H(X)$, $H(Y)$ are the marginal entropies of X and Y respectively.

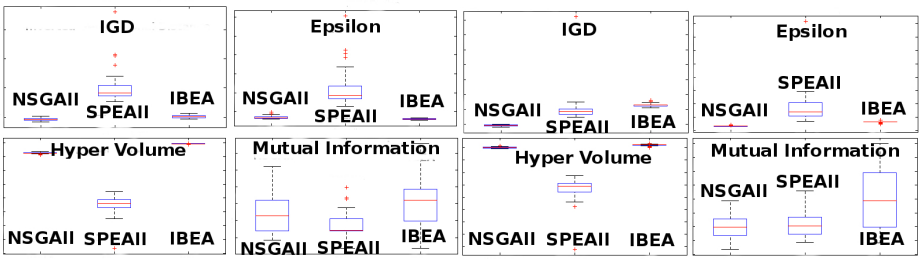
The way mutual information is applied would differ depending on the independence assumptions among the objectives. The definition in Sec3.1 is only valid for univariate random variables, so if we work on the assumption that the objectives are all independent then mutual information can be measured separately between the approximated PF and the true PF, one objective at a time and then the mutual information indicator is defined as: $MI(A,B) = \sum_{i=1}^{n} \alpha_i U(A_i, B_i)$, where A is the approximated PF, B is the true PF, A_i is the values of objective i from the PF A, n is the number of objectives, and α is the weight vector where $\alpha_i \leq 0$ and $\sum_{i=1}^{n} \alpha = 1$. The result of this indicator quantifies how much the approximated PF reduces uncertainty about the true PF. The higher MI then the better approximation A is to the true PF.

This is a simple and reliable measure that can be applied to any MOP regardless of the shape of the PF. It does not require a reference point and does not depend on one aspect of the data like spread or diversity. Another major drawback of this indicator is that the histogram estimation would allow two unequal PFs to produce the same MI if they have similar histograms, as demonstrated in Figure 1a. To circumvent this problem a novel method for mutual-information based indicator is proposed in the next section.

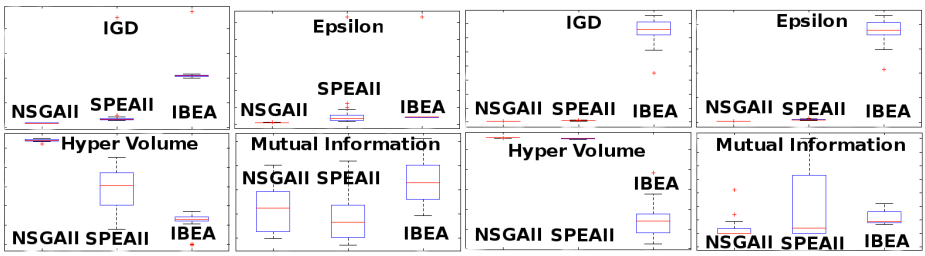


(a) (b)

Fig. 1. (a) An example of two unequal PFs with the same histogram. The blue PF (A) is the true PF for ZDT1 and the red PF (B) is a shifted copy of the blue PF. MI(A,A) = MI(A,B) = 1. (b) The blue dots belong to a hypothetical true PF. The red dots belong to a hypothetical approximated PF. The dots within the circle belong to the approximated PF and are considered outliers.



(a) ZDT1 (b) ZDT2



(c) ZDT3 (d) ZDT4

Fig. 2. Results for ZDT1-4 using the three algorithm and the four quality indicators.

3.2 Processing Pareto Fronts as Images

The main idea here is to create a pixel-based image for each PF, and then calculate mutual information among these newly created images. In order for these images to be comparable they should all have the same resolution per dimension, have the same origin and the same image size.

Given iA: the image of the approximated PF A and iB: the image of the true PF B then a new indicator is defined as: $I_{MI(A,B)} = U(iA, iB)$.

The highest resolution possible for the images is the minimum difference between two adjacent points of all the PF of interest which might be a very high resolution that the generated images would be extremely large to be practical for processing, so the resolution can be reduced in order to generate images of reasonable size. There must be a careful balance here as low resolution images could lead to a large lose of information affecting the quality of the measure itself. Following are the steps to create the PF images. 1) rescale all PFs so they all have the same origin 2) calculate the ratio between the required resolution and the maximum objective value among all PFs which is necessary to project from objective space to the image pixel space. 3) each point in PF is transated to a pixel of value 1 in the resulted image and the rest are zeros.

Going back to the problem of unequal PFs with equal histograms. If we used the new measure as an indicator in Fig.1a then $I_{MI}(A, A) = 1$ and $I_{MI}(A, B) = 0.0018$ which clearly shows that the problem is resolved.

3.3 Handling Outliers

One of the main features of any quality indicator is its ability to handle outliers in the approximated PF. Outliers can cause a bias in the quality measure calculation especially if a distance measure is used. Some measures try to reduce this effect, for example I_{IGD} is calculated by starting from the true PF and then trying to find the closest points in the approximated PF. Although this minimizes the effect of outliers, it still does not provide a fair comparison as it will give the same value for an approximated PF with or without the outliers.

Figure 1b demonstrates this effect . In this example we created a hypothetical true PF and approximated PF that contains some outliers. To check the robustness of the quality indicators we calculate the value of the indicator using the approximated PF with and without the outliers. The results show that I_{MI} and I_{hv} are robust to the outliers and are able to distinguish them (I_{hv}=0.2284 and 0.2332 without and with the indicators respectively and I_{MI} =0.0008 and 0.0007) while the other two indicators could not (I_{IGD}=0.0145 for both cases and I_{ϵ}=0.3911 for both cases as well).

4 Experiments

It is usually tricky to evaluate the evaluation metric. Here we tackle this issue indirectly by comparing the results of three other quality measures and discuss

how each, and I_{MI}, handle different cases of evaluation. To test the newly developed measure 5 standard two-dimensional MOPs are used: ZDT1-ZDT4, and ZDT6 [13]. The selected test problems cover diverse MOPs with convex, concave, connected and disconnected PFs. These problems were frequently used to verify the performance of several algorithms in the field of multi-objective optimisation.

The test MOPs are solved using three popular multi-objective optimisation methods: NSGAII [14], SPEAII [15], IBEA [16].

For each MOP the three methods were run 30 times using a population of 300 individual and lasted for 250 generations. The quality of the approximated PFs from the five MOPs is measured using four indicators: I_{IGD}, I_{ϵ}, I_{hv}, and I_{MI}. For I_{MI} calculation a resolution is set to $1000X1000$.

5 Results and Discussion

To demonstrate the results of the approximated PF using the three methods and tested by the four indicators, for each MOP four plots are generated each for each indicator. Each of these plots contains a box plot representation of the values of one indicator applied on the 30 runs for each of the three methods, Figures 2a - 2d with results of ZDT6 not presented due to limited space.

Looking carefully at the figures, the different indicators draw a rather vague, and somehow confusing, picture about the performance of the different methods. Although most of the differences among the different methods are statistically significant they do not always go in the same direction for different indicators. This is a known issue in the quality assessment of mutliobjective optimizers via indicators [5]. This is interpreted as different indicators provide different information regarding the approximated PF so one can chose the optimizer based on what is more relevant to the application.

The I_{MI} indicator seems to be giving a slightly different view from the rest of the indicators. For instance, it is the only indicator to show no significant difference in some cases which actually reflects more what we see from visual inspection of the approximated PFs. It also usually shows less variance among the different runs and fewer outliers which can be credited to its probabilistic nature.

By definition mutual information is only applied on univariate random variables and hence an independence assumption is imposed among the objectives. If the objectives are dependent, which would be the case in most problems, then the previous definition of mutual information is not accurate. However, by transforming the PF to an image all the dependency information are preserved and hence the indicator I_{MI} can be seen as a multivariant approach.

Because the mutual information function uses estimated probability density functions/ distributions, it is less affected by outliers and can produce a much more robust results than the indicators that use distance functions.

References

1. Talbi, E.: Metaheuristics - From Design to Implementation. Wiley (2009)
2. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Genetic and Evolutionary Computation. Springer, New York (2007)
3. Deb, K.: Multi-objective optimization. In: Multi-Objective Optimization using Evolutionary Algorithms, pp. 13–46 (2001)
4. Al Moubayed, N., Hasan, B.A.S., Gan, J.Q., Petrovski, A., McCall, J.: Continuous presentation for multi-objective channel selection in brain-computer interfaces. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. IEEE (2012)
5. Fonseca, C.M., Knowles, J.D., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. In: Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), vol. 216 (2005)
6. Fonseca, C.M., da Fonseca, V.G., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello, C.A.C., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 250–264. Springer, Heidelberg (2005)
7. Grunert da Fonseca, V., Fonseca, C.M., Hall, A.O.: Inferential performance assessment of stochastic optimisers and the attainment function. In: Zitzler, E., Deb, K., Thiele, L., Coello, C.A.C., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 213–225. Springer, Heidelberg (2001)
8. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithm research: A history and analysis. Air Force Inst. Technol., Dayton, OH. Tech. Rep. TR-98-03 (1998)
9. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
10. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
11. Scott, D.W.: On optimal and data-based histograms. Biometrika 66(3), 605–610 (1979)
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization, pp. 105–145 (2005)
14. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: Nsga-ii. IEEE Trans. on Evolutionary Computation 6(2), 181–197 (2002)
15. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm (2001)
16. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)

Velocity Divergence of CCPSO in Large Scale Global Optimization

Shanqing Hu and Bin Li

School of Information Science and Technology,
University of Science and Technology of China, Heifei, Anhui, China
hushanq@mail.ustc.edu.cn, binli@ustc.edu.cn

Abstract. Cooperative co-evolution particle swarm optimization (CCPSO) has been testified to be a promising algorithm for large scale global optimization problems(LSGOs). In this paper, a new side about the effect of velocity in CCPSO is investigated. A phenomenon has been found that some dimensions' velocities in particles refuse to shrink when the values of these dimensions in real global optimal solutions are close to the bound. The divergence of velocity will decrease the convergence speed of algorithm on these dimensions, and therefore decrease the final quality of the obtained best solution. To deal with this problem, a velocity divergence treatment method is proposed, and a new variant of CCPSO, called VD-CCPSO, is presented to improve its performance on LSGOs. Experiment results testify the effectiveness of the proposed method.

Keywords: Cooperative Co-evolution, Particle Swarm Optimization, Large-Scale Global Optimization, Velocity Divergence, Velocity Re-initialization.

1 Introduction

Since its birth in 1995[1][2], particle swarm optimization (PSO) has been widely studied and successfully applied. As a population-based optimization technique, PSO also faces the challenge of balancing the capability of local convergence and global exploration. To get good balance, different topology structures [3][4][5][6], parameters controlling strategies[7][8] , velocity initialization and re-initialization methods [5][9][10] have been introduced into original PSO.

Like many other Evolutionary Algorithms (EAs), conventional PSO also suffers from the curse-of-dimensionality. With the dimensionality of the search space increasing, the performance of conventional PSO prones to deteriorate dramatically. Cooperative Coevolution(CC) is a effective technique to tackle the curse-of-dimensionality problems. Many successful cases of incorporating CC into conventional EAs to increase their performance on LSGOs have been reported[11][12][13]. The first attempt of introducing CC into PSO was done by Van den Bergh and Engelbrecht [14] where a static grouping strategy was adopted, and its effectiveness was tested on functions up to 30 dimensions [14]. Li et. Al. proposed two new cooperative coevolving particle swarm optimization CCPSO and CCPSO2[15][6],

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 545–552, 2013.

aiming to improve the performance of PSO on LSGOs. In CCPSO, *random grouping* was adopted.

When using CCPSO with *random grouping* to deal with high-dimension problems, an interesting phenomenon was found, that is, for some LSGOs whose real global optimal solution is close to the border of search space, at the later period of searching process, some dimensions' velocities of particles refuse to shrink when the values of these dimensions in real global optimal solution are close to the bound. The large velocity is detrimental in the later period of the optimization course. To deal with such problem, a new velocity divergence detection method is proposed, and a re-initialization strategy is adopted as a makeup approach. With these techniques, a new variant of CCPSO, VD-CCPSO, is presented to improve its performance on LSGOs whose real global optimal solution is located near the bound. Experiment results testify the effectiveness of the proposed method.

The rest of this paper is organized as follows: section 2 discusses the related works. Section 3 shows the velocity divergence phenomenon of CCPSO on LSGOs, and presents the method to tackle the problem. Experiment setting and results are given in Section 4. Finally, section 5 concludes the whole work presented in this paper.

2 Related Works

Velocity is an important element in PSO which drives particles flying in the search space, it gravely influence the exploitation and exploration property of algorithms. It is commonly acknowledged that the velocity size, absolute value of velocity, direct influences the step size of particles and determines the search granularity. Large velocity size in the beginning of search is favorable for exploration to find more promising region, while smaller velocity size is more suitable in the later searching period to contact more fine-grained search.

Velocity initialization strategy is also a very important element for PSO [10]. There have been three initialization stragies:1) Initializing velocities to zero, 2)initializing velocities to random values within the domain of the optimization problem, 3)Initializing velocities to small random values. The last two strategies are often the best choice.

In order to prevent step sizes from becoming too large or too small, adding clamp to velocity is necessary. A linearly decreasing V*max* is introduced in [16]. Generally, V*max* can be set to δ*(-X*max*, X*max*) and the variation of δ depends on the optimization problems [17]. It also should be noticed that velocity clamp influences not only the search steps but also the directions of particles.

To monitor the velocity and keep swarm diversity to avoid trapping into local optima is also necessary [9]. When the mean norms of the swarm velocity is smaller than the predefined stagnation threshold, the swarm is considered to be stagnate, and a re-initialization strategy usually be adopted to help the swarm jumping out of possible local optimal.

3 Velocity Divergence of CCPSO

In the end of an optimization process, smaller velocity would be helpful to fine turning and convergence of the algorithm. When studying the behavior of CCPSO with random grouping on LSGOs, we found an interesting phenomenon. In this section, the phenomenon, velocity divergence in the later searching process, will be depicted, and a method will be introduced to deal with the bad effect caused by the velocity divergence.

3.1 CCPSO

In early CCPSO-S_k [15], the n-dimensional search space is decomposed into K s-dimensional subspaces (where $n = K*s$), each subspace has one swam of s dimensions search in it; The group number is unchanged though the course of optimization. A random grouping strategy is adopted that chooses group size randomly from a size pool before each iteration. Algorithm 1 shows the pseudocode of CCPSO.

1) Create and initialize NP particles positions and velocities
2) randomly choose group size s from a set S and let K=n/s,
3) randomly permutate all n dimensions indices and according the indices construct K swarms, each contain s dimensions.
4) for each group j=1:K do
for each particle i=1:NP
update particle velocity, Vi = min(Vmax,max(Vmin,Vi)
update particle position and calculate fitness value
if fitness value better than pbest fitness value, update pbest fitness value and position
if fitness value better than gbest fitness value, update gbest fitness value and position
end
end
5) stop iteration if termination condition is met; otherwise goto step 2)

Algorithm 1. The pseudocode of the CCPSO algorithm

3.2 Velocity Divergence of CCPSO

In order to understanding the velocity divergence of CCPSO, particle velocities of all dimensions should be recorded. A matrix **VM**, shown in Eq.1, is introduced for this purpose, in which V_{ij} recoards the mean of absolute velocities of all particles on j^{th} dimension in i^{th} iteration

$$\mathrm{VM}=\begin{bmatrix} V_{1,2} & V_{1,2} & \cdots & V_{1,w} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,w} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m,1} & V_{m,2} & \cdots & V_{m,w} \end{bmatrix} \tag{1}$$

The CCPSO with random grouping is executed on 20 test functions for CEC 2012 LSGO Competition. The velocity of some representative dimensions, dimensions with velocity divergence and dimensions with velocity convergence, are recorded with the VM defined in Eq.1. By plotting the column of VM, the variation of the corresponding dimension's velocity along with iteration can be depicted and compared with each other.

To investigate the behavior of dimensions with different closeness to the bound, a definition of closeness is presented first. Set $\Delta_j = (ub_j\text{-}lb_j)/200$, where ub_j and lb_j donate the upper bound and lower bound of the jth dimension. Let δ represents the set of dimensions whose values in real global optimal solution fall in the region between [ub-Δ, ub] or [lb, lb+Δ], then let 2δ stands for the set of dimensions whose values in real global optimal solution fall in the region [ub-2Δ, ub-Δ] or [lb+Δ, lb+2Δ], and 3δ in the region [ub-3Δ, ub-2Δ] or [lb+2Δ, lb+3Δ], and so on. Those dimensions belong to set δ, 2δ and 3δ are called *border dimensions.* For example, if the valid region of all dimensions of F1 is (-100,100), 2δ represents the dimensions whose values in real global optimal solution fall in the region [98, 99] or [-99, -98]. Besides *border dimensions*, the velocity of some representative *normal dimensions* are also recorded and depicted. The *normal dimensions* are those whose values in real global optimal solution are far from the border, i.e. beyond 5δ.

Fig.1 shows the variation of velocities of the representative dimensions of both border dimensions and normal dimensions on 20 test functions. In Fig.1, the curves denoted as 1δ, 2δ and 3δ, are the mean values of all dimensions belonging to the set δ, 2δ and 3δ respectively. And the curve denoted as *normal* is the mean value of 10 randomly selected dimensions belonging to the set *normal dimensions*. It can be seen that almost all *border dimensions* on all test functions show, more or less, the velocity divergence phenomenon, while the velocities of *normal dimensions* converge to 0 as expected, except for F04.

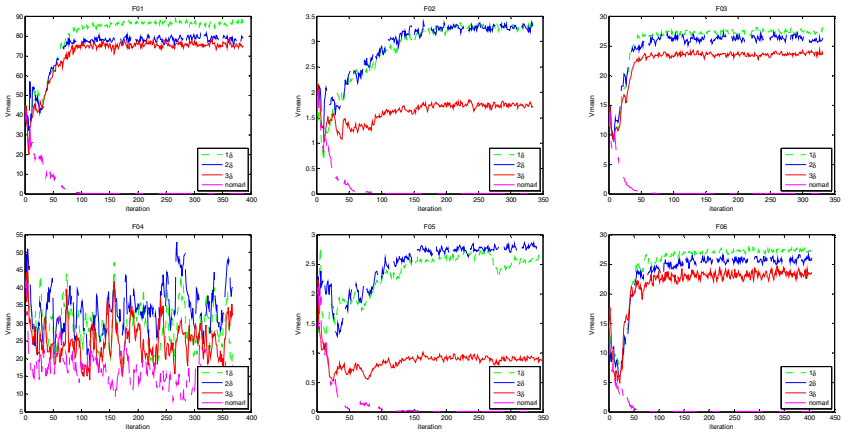


Fig. 1. Velocity comparison between border dimensions and normal dimensions on 20 LSGO test functions

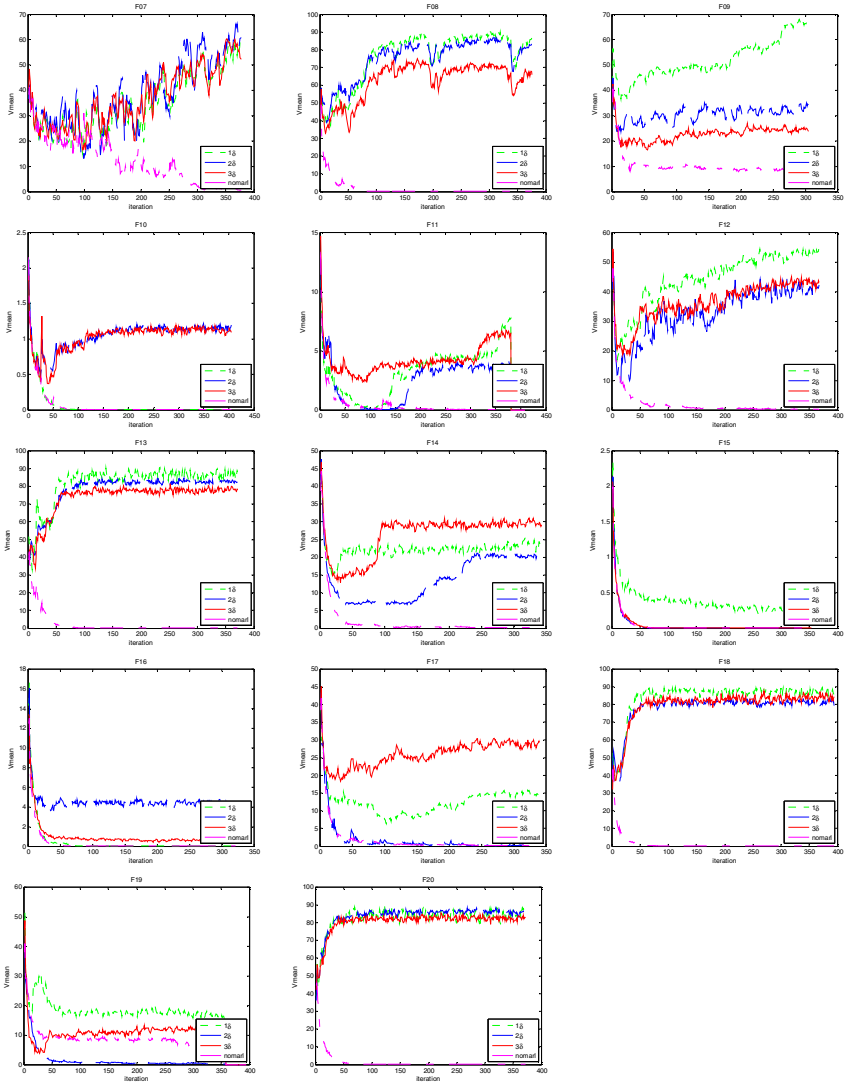


Fig. 1. (*continued*)

3.3 Method to Deal with Velocity Divergence

In early stage of optimization, large velocities can drive particles to search a wider area and keep population diversity, so we do not need concern velocity divergence. But with the evolution get into the later period, large velocity is not expected, which will prevent algorithm from getting to the global optimal efficiently. Therefore, some method needs to be presented to deal with the velocity divergence, and to improve the performance of CCPSO with random grouping on LSGOs in the later period of evolution. The method presented in this paper is briefly introduced as below.

1) Calculate the probability of conducting velocity divergence treatment.
Such probability should increase with the increase of "evolution time". In this paper, the probability is set to increase linearly with the increase of the number of fitness evaluations, as shown in Eq.2.

$$\mathrm{VP}=k*FEs/Max\_FEs \tag{2}$$

where k = 0.6 donates the increase rate of VP. *FEs* represents the current fitness evaluation size, and *Max_FEs* =3.0E+06 stands for the maximum fitness evaluation number available for the optimization task.

2) Dimension selection
When VP > 0.2, which means Fes > 1.0E+06, the velocity divergence treatment can be started. For each iteration (e.g. *i*th) of CCPSO, a random number *r,* uniformly distributed in [0,1], is generated. If *r* < VP, the velocities in *i*th row of VM are sorted decreasingly, 20 dimensions with largest velocity values are selected.

3) Velocity re-initialization and optimization of selected dimensions
The velocities of selected dimensions are re-initialized to zeros. After that, the selected 20 dimensions are put into one group and optimized using standard PSO. When VP>0.5, the standard PSO is modified by updating velocities only with *gbest* to enhance the convergence.

The CCPSO with Velocity Divergence treatment is named VD-CCPSO in this paper.

4 Experiment Results

The parameter setting of CCPSO and VD-CCPSO are same for the experiments. The population size *NP* is set to 30, maximum fitness evaluation number *Max_FEs*=3000000, inertia weight *w*=0.729, acceleration factor c1=c2=1.49, a group size set S={2, 5, 10, 50, 100, 250} is used for random selecting the group size in random grouping strategy[6].

Five test functions presented in CEC 2012 LSGO Competition are adopted benchmarks: *f1(ShiftedSphere)*, *f2(ShiftedRosenbrock), f3(ShiftedGriewank), f4(ShiftedSchwefel), f5(ShiftedElliptic)*. All the experiment results are the statistic of 25 independent runs of the algorithms.

Table 1 shows the experiment results of CCPSO and VD-CCPSO on the test functions.

In table 1, *BDS* stands for the number of dimensions close the border. It can be seen, for *BDS* = 50, 200 and 500, VD-CCPSO have obviously better performance than CCPSO on *f1*,*f3* and *f5*. While on *f2* and *f4,* VD-CCPSO does not show obvious advantage. For *BDS* = 1000, which means all dimensions are shifted close to the border, VD-CCPSO get some better solutions on *f2* and *f4*. These results show the effectiveness of consideration of velocity divergence.

It can also be seen that, with the increase of *BDS*, the performance of both algorithms are getting bad. That probably means the more dimensions in the real global optimal solution close to the border, the harder the test function to be optimized.

Table 1. The experiment results of CCPSO and VD-CCPSO on test functions

Algorithm	**Quality**	*f1*	*f2*	*f3*	*f4*	*f5*	*f1*	*f2*	*f3*	*f4*	*f5*
BDS		*50*					*200*				
CCPSO	Best	1.54E-03	1.44E+03	1.61E-06	4.51E+02	2.98E+01	2.70E-02	1.66E+03	4.16E-05	4.34E+02	5.55E+02
	Median	3.78E-03	1.80E+03	8.98E-06	7.74E+02	1.73E+02	3.64E-02	2.14E+03	9.23E-05	7.39E+02	1.30E+03
	Worst	8.60E-03	2.18E+03	3.68E-02	1.80E+03	1.11E+03	5.41E-02	2.44E+03	1.73E-02	1.63E+03	2.77E+03
	Mean	3.96E-03	1.80E+03	4.63E-03	8.48E+02	2.89E+02	3.70E-02	2.13E+03	1.18E-03	8.67E+02	1.36E+03
	Std	1.73E-03	1.91E+02	8.58E-03	3.54E+02	2.57E+02	8.37E-03	1.90E+02	3.89E-03	3.74E+02	5.75E+02
VD-CCPSO	Best	**1.61E-26**	1.30E+03	**3.77E-15**	3.62E+02	**8.10E-21**	**5.01E-09**	1.79E+03	**1.47E-08**	3.10E+02	**1.22E-09**
	Median	**2.59E-22**	1.65E+03	**4.11E-15**	6.82E+02	**3.60E-19**	**9.04E-05**	2.22E+03	**1.42E-06**	8.29E+02	**7.31E-06**
	Worst	**7.43E-17**	5.30E+03	**2.71E-02**	1.87E+03	**1.03E-15**	**1.29E-03**	7.68E+03	**1.97E-02**	1.74E+03	**5.44E-02**
	Mean	**2.97E-18**	1.77E+03	**2.66E-03**	7.65E+02	**5.77E-17**	**1.92E-04**	2.60E+03	**2.37E-03**	8.51E+02	**2.94E-03**
	Std	**1.49E-17**	7.62E+02	**6.31E-03**	3.29E+02	**2.09E-16**	**2.77E-04**	1.44E+03	**5.29E-03**	3.57E+02	**1.12E-02**
Algorithm	**Quality**	*f1*	*f2*	*f3*	*f4*	*f5*	*f1*	*f2*	*f3*	*f4*	*f5*
BDS		*500*					*1000*				
CCPSO	Best	1.96E-01	2.77E+03	3.98E-04	3.47E+02	4.45E+03	2.19E+01	1.04E+04	4.26E-02	3.70E+02	6.41E+04
	Median	3.85E-01	3.44E+03	9.77E-04	6.68E+02	9.00E+03	5.22E+01	1.54E+04	9.21E-02	7.22E+02	9.41E+04
	Worst	5.80E-01	1.10E+04	2.02E-02	1.58E+03	1.34E+04	6.93E+01	1.29E+05	1.69E-01	1.14E+03	1.39E+05
	Mean	4.02E-01	5.01E+03	3.72E-03	7.00E+02	9.18E+03	4.87E+01	2.03E+04	1.00E-01	7.09E+02	9.71E+04
	Std	1.02E-01	2.82E+03	5.27E-03	2.79E+02	2.37E+03	1.53E+01	2.29E+04	3.59E-02	2.74E+02	**1.99E+04**
VD-CCPSO	Best	**6.50E-03**	2.34E+03	**2.36E-05**	3.00E+02	**3.61E+01**	1.35E+01	**7.92E+03**	3.56E-02	3.31E+02	**1.67E+04**
	Median	**1.97E-02**	3.08E+03	**7.32E-05**	7.18E+02	**1.68E+02**	3.15E+01	**1.14E+04**	6.66E-02	6.71E+02	**2.28E+04**
	Worst	**5.67E-02**	1.37E+04	**1.24E-02**	1.79E+03	**1.07E+03**	3.10E+02	**1.10E+05**	1.26E-01	2.32E+03	**3.91E+04**
	Mean	**2.19E-02**	5.58E+03	**2.74E-03**	8.36E+02	**3.05E+02**	4.76E+01	**1.68E+04**	6.98E-02	8.19E+02	**2.49E+04**
	Std	**1.34E-02**	3.99E+03	**4.47E-03**	3.58E+02	**3.07E+02**	5.73E+01	**2.03E+04**	2.50E-02	4.91E+02	**6.73E+03**

5 Conclusions

In this paper, the velocity divergence phenomenon of CCPSO is depicted on high dimensional optimization tasks. A method is proposed to deal with the bad effect of the velocity divergence. With this method, a new variant of CCPSO, VD-CCPSO, is presented. The effectiveness of the proposed method is testified by the experiment results on test functions from CEC 2012 LSGO Competition.

References

1. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
3. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proc. IEEE Congr. Evol. Comput., vol. 2, pp. 1671–1676 (May 2002)
4. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. IEEE Trans. Evol. Comput. 8(3), 204–210 (2004)
5. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particles swarm optimization for global optimization of multimodal functions. IEEE Trans. Evol. Comput. 10(3), 281–295 (2006)
6. Li, X., Yao, X.: Cooperatively Coevolving Particle Swarms for Large Scale Optimization. IEEE Trans. Evol. Comput. 16(2), 210–224 (2011)
7. Zhan, Z.-H., Zhang, J., Li, Y., Chung, H.S.-H.: Adaptive particle swarm optimization. IEEE Trans. Syst. Man Cybern. B 39(6), 1362–1381 (2009)
8. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6(1), 58–73 (2002)

9. Binkley, K.J., Hagiwara, M.: Balancing Exploitation and Exploration in Particle Swarm Optimization Velocity-based Reinitialization. Information and Media Technologies 3(1), 103–111 (2008)
10. Engelbrecht, A.: Particle Swarm Optimization Velocity Initialization. In: Proc. IEEE WCCI, June 10-15, pp. 1–8 (2012)
11. Potter, M., Jong, K.D.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994)
12. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Inform. Sci. 178(15), 2986–2999 (2008)
13. Zhang, K., Li, B., Tan. L.: Empirical study of the effect of variable correlation on grouping in Cooperative Coevolutionary Evolutionary Algorithms. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. IEEE (2012)
14. van den Bergh, F., Engelbrecht, A.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. 8(3), 225–239 (2004)
15. Li, X., Yao, X.: Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms. In: Proc. IEEE CEC, pp. 1546–1553 (May 2009)
16. Fan, H.Y., Shi, Y.: Study on Vmax of particle swarm optimization. In: Proc. Workshop Particle Swarm Optimization, Indianapolis, IN (2001)
17. Shi, Y., Eberhart, R.C.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
18. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.-P., Chen, C.-M., Yang, Z.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large Scale Global Optimization. University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, China. Tech. Rep (2010)

Machine Learning Enhanced Multi-Objective Evolutionary Algorithm Based on Decomposition

Yung Siang Liau, Kay Chen Tan, Jun Hu, Xin Qiu, and Sen Bong Gee

Department of Electrical and Computer Engineering
National University of Singapore, Singapore
{liauys,eletankc,junhu,qiuxin,a0039834}@nus.edu.sg
http://www.nus.edu.sg

Abstract. We address the problem of expensive multi-objective optimization using a machine learning assisted model of evolutionary computation. Specifically, we formulate a meta-objective function tailored to the framework of MOEA/D, which can be solved by means of supervised regression learning using the Support Vector Machine (SVM) algorithm. The learned model constitutes the knowledge which can be then utilized to guide the evolution process within MOEA/D so as to reach better regions in the search space more quickly. Simulation results on a variety of benchmark problems show that the machine-learning enhanced MOEA/D is able to obtain better estimation of Pareto fronts when the allowed computational budget, measured in terms of number of objective function evaluation, is scarce.

Keywords: Evolutionary multi-objective optimization, support vector machine, expensive optimization, surrogate modelling.

1 Introduction

A multi-objective optimization problem [3] can be formulated as follows:

$$\begin{aligned} &\min \boldsymbol{f}(\boldsymbol{x}) = \left[f_1(\boldsymbol{x})\ f_2(\boldsymbol{x}) \ldots f_m(\boldsymbol{x})\right] \\ &\text{subject to } \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where m is the number of objectives to be considered, $\boldsymbol{x} \in \mathbb{R}^d$ is a decision vector belonging to the m-dimensional decision space, and $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^m$ is a mapping from the decision space to the objective space.

We are interested in improving the capabilities of current multi-objective evolutionary algorithms in solving *expensive* optimization problems [10]. Expensive optimization problems arise frequently in real-world scenario, in which the objective function $\boldsymbol{f}$ is very time consuming to evaluate, due to factors such as the involvement of high-fidelity computer simulation. We attempt to tackle the problem by introducing a technique based on machine learning that can help accelerate the evolutionary search performed by evolutionary multi-objective

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 553–560, 2013.

optimization (EMO) algorithms – to be able to achieve a set of solution with better quality, under a stringent computational budget.

Our contributions are listed as follow: (i) We formulate a supervised learning problem under the decomposition based framework of EMO, whose solution encodes knowledge that could provide guidance for the evolutionary operators when evolving new solutions, thereby speeding up the search process. To the best of our knowledge, previous related works in the literature focus on the domination based framework; (ii) we propose the use of Support Vector Machine (SVM) based regression method to solve for the learning problem; (iii) the learning framework that is introduced can be used to augment general decomposition based EMO algorithms, while in this paper we show a specific example of its use tailored to MOEA/D.

1.1 Related Work

Surrogate modelling assisted evolutionary computation is a paradigm that tackles the same problem of expensive optimization [5]. In order to extract knowledge useful for guiding the evolution, a data mining approached has been successfully implemented to cope with noisy optimization problem [2]. In [9], the authors proposed a method to directly learn to predict the Pareto rank of the individuals, given their decision variables using ranking-based SVM. In [7], a Pareto surrogate model based SVM is also formulated for enhancing multi-objective optimization. It is noted that these works largely focus on Pareto-based framework. Our proposed approach is tailored to its counterpart – the decomposition-based framework. Our approach could benefit the class of problems on which a decomposition-based framework is more suitable.

2 A Meta-objective Formulation

2.1 Background on MOEA/D

Multi-objective evolutionary optimization algorithm based on decomposition (MOEA/D) [11] is a recently proposed algorithm that is capable of handling optimization problems with complicated Pareto set well [6]. A possible decomposition approach is via the Tchebycheff method, where the decomposed optimization problem can be stated as [11]

$$\begin{aligned} &\min g^{te}(x|\lambda, z^*, f) = \max_{1\leq i\leq m} \{\lambda_i |f_i(x) - z_i^*|\} \\ &\text{subject to } x \in \Omega \end{aligned} \tag{2}$$

whose optimal solution is also a Pareto optimal solution of the original multi-objective problem. In MOEA/D, each individual is associated with a unique weight vector λ^p that defines a unique single objective optimization problem in the form of E.q. (2), which is to be optimized. A new individual will replace an old individual in case the new individual turns out to be a better solution to the associated scalarized objective function than the old individual.

2.2 Formulation of the Learning Task

We propose that E.q. (2) can be *predicted* to a certain degree of accuracy via a machine learning approach. From a high level perspective, as the evolution progresses, individuals associated with different weight vectors along with their evaluated fitness vectors collectively form a dynamic data-set that could be used to approximate the fitness function. We want to have a learning model that could help answer the question: *Given an individual x with unknown objective values to be determined by f, how good will it likely to perform with respect to a sub-function with a weight vector λ, when the current state of an approximated ideal vector, z^* is observed?*

As E.q. (2) is dependent upon the weight vectors, we first define a meta-objective function $\mathcal{G} : \mathbb{R}^{m+d} \rightarrow \mathbb{R}$ which will be estimated by a learning algorithm. Formally, it can be expressed as

$$\mathcal{G}(\lambda, x|z^*, f) = g^{te}(x|\lambda, z^*, f) \tag{3}$$

where the domain of $\mathcal{G}$ is extended to include λ and x simultaneously compared to that of g. By the definition of this meta-objective function, it encodes the answer to the question that we pose previously. This is clearly a regression learning problem.

Importantly, learning a single meta-function as defined by E.q. (3) makes it possible for a learning algorithm to leverage the implicit relationship between the decision vectors, the weight vectors and different sub-functions simultaneously. In the next section, we will describe in details the algorithmic steps on the on-line construction of the training data set and how this knowledge can be incorporated during the evolution.

2.3 Support Vector Regression (SVR)

Support vector Machine (SVM) is one of the most popular class of algorithms in the machine learning community due to its superior generalization ability. This motivates our choice of using SVM in solving the regression learning task that will be detailed in the following section. The specific class of SVM algorithm that will be utilized in our proposed MOEA enhancement is the ν-Support Vector Regression (ν-SVR) [8]. The ν-SVR solves the following optimization problem:

$$\begin{aligned} &\min_{w,b,\xi,\xi^*,\epsilon} \frac{1}{2} w^T w + C \left(\nu\epsilon + \frac{1}{l} \sum_{i=1}^{l} (\xi_i + \xi_i^*) \right) \\ &\text{subject to} \begin{cases} (w^T \phi(x_i) + b) - z_i \leq \epsilon + \xi_i \\ z_i - (w^T \phi(x_i) + b) \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l \\ \epsilon \geq 0 \end{cases} \end{aligned} \tag{4}$$

where $\phi(x_i)$ is a mapping from the input space to the feature space that can be efficiently computed using the kernel trick due to the structure of the dual

formulation of E.q. (4). Due to space constraints, the dual formulation is omitted here and interested readers are encouraged to refer to a more detailed treatment in [8]. In the context of estimating the function $\mathcal{G}$ in E.q. (3), the resulting SVR estimation which we will denote as $\hat{\mathcal{G}}$ can be written as

$$\hat{\mathcal{G}}\left(\lambda, x | z^*, f\right) = \sum_{i=1}^{l} \left(-\alpha_{i|z^*,f} + \alpha^*_{i|z^*,f}\right) K\left(x_i, \left[\lambda, x\right]\right) + b_{z^*,f} \tag{5}$$

where $\alpha_{i|z^*,f}$'s are the optimized Langrangian multipliers given the data set computed based upon the current state of the evolution encoded in the tuple (z^*, f), and K is the kernel mapping.

3 A Machine-Learning Enhanced MOEA/D

We described the complete algorithm, which we will refer to as SVM-MOEA/D in Algorithm 1. It should be noted that no modification at all is made to the original replacement strategy used by the MOEA/D.

3.1 On-line Generation of Training Data

Data Construction. The meta objective function $\mathcal{G}$ will be estimated through the learning algorithm, which will require a training data set. In our approach, the training data set will be constructed by collecting and accumulating data points, as new individuals are being generated and evaluated. Each training data point corresponds to a single individual. The feature vector will be formed by concatenating the associated weight vector and decision variable. Then, the target value for that particular data point is computed using Eq. (3). We will refer to this training data as *training population*.

Data Updating. Due to computational limitation, training population cannot grow indefinitely. Here, we adopt a simple approach of retaining only the newest data point, essentially yielding a first-in-first-out (FIFO) database. This simple approach is empirically observed to be effective as the domain of unknown test sample (i.e. unevaluated individuals) via $\mathcal{G}$ is naturally restricted (by the evolutionary operators) to be near to the current pool of parent populations, rendering the information from very old parents less useful. Each time before the training of the SVR model, the target values of all the entries in the training population will be updated using E.q. (3), importantly with the most updated approximation of the ideal vector z^*.

3.2 Model-Based Informed Evolution

Extended Sampling. In order to utilize the knowledge encoded in the learned model, instead of generating a *single* solution, the algorithm will generate *multiple* solutions in a list of length L_s without immediately evaluating their objective

values. This extended sampling procedure can be seen as a way to provide additional chances that the algorithm could come across good solution, given it's current evolutionary operators. The problem then boils down the determining which candidate solution to choose for evaluation, which will then enter the population to compete with the existing solutions.

Model-Based Filtering. During the individual replacement phase in MOEA/D, a candidate solution would be competing against either a subset of the subproblems (defined by the neighbourhood table) or simply against all the subproblems. To obtain a quality measure for each of the candidate solution x (with unknown objective values), two scenarios are discussed separately as follows, each addressing one of the above replacement types:

1. Replace within $\mathcal{N}$-th neighbourhood: Denoting the set of weight vectors associated with neighbourhood as $\Lambda_{\mathcal{N}} = \{\lambda_1^{\mathcal{N}}, \lambda_2^{\mathcal{N}}, \ldots, \lambda_{|\Lambda_{\mathcal{N}}|}^{\mathcal{N}}\}$, the quality of x *w.r.t* this neighbourhood, denoted as $\mathcal{P}(x)$, is obtained from the *averaged* prediction output of $\hat{\mathcal{G}}\left([\lambda_i, x]\right)$ over $i = \{1, 2, \ldots, |\Lambda|\}$.
2. Replace the whole population: A good quality for x demands the ability of x to be able to replace as much of the individual in the population as possible. Thus, the quality estimate $\mathcal{P}(x)$ in this case can be obtained using the same approach as described above, by simply changing the neighbourhood to be the whole population, namely $\Lambda' = \{\lambda_1, \lambda_2, \ldots, \lambda_{|Pop|}\}$.

To summarize, given a pool of potential candidates $\mathcal{C} = \{x_1, x_2, \ldots, x_{|C|}\}$, model-based filtering using the learned model $\hat{\mathcal{G}}$ can be obtained by extracting the index associated with the individual x with the best predicted quality $\mathcal{P}(x)$, via $c_{\text{best}} = \text{argmin}_{k \in \{1,\ldots,|C|\}} \mathcal{P}(x_k)$, where $x_{c_{best}}$ is the chosen individual to be evaluated and to enter the evolution with the current population.

4 Experimental Results

4.1 Experiment Settings

We base the implementation of our SVM-MOEAD upon the source code of MOEA/D provided in [6] and the SVM routines in [1]. The population size used is 100 for all the algorithms. The parameter ν is set as the default value 0.5, and a stump kernel is used during the training of SVM. The considered benchmark problems include the ZDT [13], WFG [4] and UF [12] test suites.

We investigate the effects of the proposed learning mechanism when being implemented in MOEA/D with a differential evolution operator (MOEA/D-DE) and with simulated binary crossover operator (MOEA/D-SBX). We refer to the their learning-based counterparts as S-MOEA/D-DE and S-MOEA/D-SBX respectively[1]. Each algorithm configuration is executed for 30 independent runs.

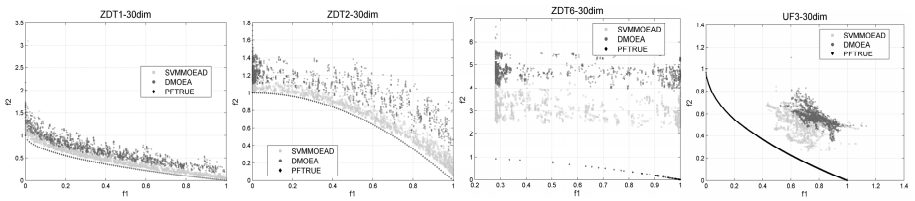
[1] It is noted that we aim to reveal the true effects of the proposed features, and thus will not perform comparison with a plethora of existing EMO algorithms, but only with the base algorithm without the learning mechanism.

Algorithm 1. SVM-MOEA/D: MOEA/D Enhanced with Informed Evolution

$W \leftarrow$ InitWeightVectors()
$NT \leftarrow$ InitNeighbourhoodTable()
$Pop \leftarrow$ InitPopulation()
$P_{\text{train}} \leftarrow \emptyset$
$N_{fes} \leftarrow 0$
while Termination condition not reached **do**
  $\hat{\mathcal{G}} \leftarrow$ UpdateModel(P_{train})
  for $i = 1$ to $PopSize$ **do**
    With probability p_{nb}, $Type \leftarrow$ Neighourbourhood
    Otherwise, $Type \leftarrow$ WholePopulation
    $Candidates \leftarrow$ ExtendedSampling(i, Pop, $Type$, NT)
    $C_{best} \leftarrow$ ModelBasedFiltering($\hat{\mathcal{G}}$, $Candidates$, i, $Type$)
    Evaluate($Candidates(C_{best})$, f)
    $N_{fes} \leftarrow N_{fes} + 1$
    ReplaceIndividuals(Pop, C_{best}, $Type$, NT, W)
    UpdateIdealVector(C_{best})
  end for
  $P_{\text{train}} \leftarrow$ UpdateTrainingData(P_{train}, Pop)
end while

4.2 Performance Analysis

The performance of the algorithms in terms of Inverted Generational Distance (IGD) [14] at 4000-th evaluation are shown in Table 1, which measures the diversity and convergence properties of an estimated Pareto front. It can be seen that under such an assumption of low allowable computational budget, the proposed variant of MOEA/D is able to obtain statistically significantly better solution in 15 and 13 benchmark problems tested respectively for DE and SBX base operators, while for the rest of the problems there is no statistically distinguishable difference in performance. In Fig. 1, we show some estimated Pareto fronts obtained with MOEA/D enhanced with the proposed learning features, for the problems on which significantly different performance metrics are observed.



(a) ZDT1 (b) ZDT2 (c) ZDT6 (d) UF3

Fig. 1. Plots of Pareto Fronts obtained over 30 independent runs

Table 1. Averaged IGD (standard deviation bracketed) values obtained by different algorithms when the number of function evaluation is fixed at 4000. Bold font implies that the algorithm is significantly better than the other using a t-test with a confidence level of 0.01. The symbols signify statistical significance of our algorithm compared to the base algorithm: ($\prec$): significantly better; ($\sim$): no significant difference; ($\succ$): significantly worse.

	(a) SBX operator		(b) DE operator	
	S-MOEAD-SBX	DMOEA-SBX	S-MOEAD-DE	DMOEA-DE
ZDT1	**0.070 (0.026)**	0.209 (0.064) ($\prec$)	**0.702 (0.150)**	1.126 (0.192) ($\prec$)
ZDT2	**0.065 (0.029)**	0.391 (0.166) ($\prec$)	**0.998 (0.173)**	2.064 (0.242) ($\prec$)
ZDT3	**0.107 (0.034)**	0.182 (0.050) ($\prec$)	**0.671 (0.132)**	0.952 (0.137) ($\prec$)
ZDT4	43.147 (9.467)	40.798 (9.444) ($\sim$)	**74.880 (13.354)**	121.531 (16.011) ($\prec$)
ZDT6	**2.427 (0.447)**	4.374 (0.384) ($\prec$)	**5.656 (0.495)**	6.778 (0.191) ($\prec$)
UF1	0.194 (0.073)	0.203 (0.067) ($\sim$)	**0.316 (0.101)**	0.407 (0.098) ($\prec$)
UF2	**0.101 (0.029)**	0.109 (0.037) ($\prec$)	0.122 (0.032)	0.119 (0.020) ($\sim$)
UF3	**0.383 (0.045)**	0.444 (0.029) ($\prec$)	**0.415 (0.055)**	0.467 (0.035) ($\prec$)
UF4	0.115 (0.006)	0.115 (0.006) ($\sim$)	0.107 (0.009)	0.105 (0.010) ($\sim$)
UF5	**1.001 (0.227)**	1.195 (0.266) ($\prec$)	**1.870 (0.398)**	2.294 (0.446) ($\prec$)
UF6	**0.603 (0.096)**	0.751 (0.153) ($\prec$)	**0.880 (0.145)**	1.584 (0.377) ($\prec$)
UF7	0.373 (0.169)	0.371 (0.139) ($\sim$)	**0.380 (0.091)**	0.454 (0.081) ($\prec$)
UF8	0.314 (0.052)	0.295 (0.051) ($\sim$)	**0.315 (0.049)**	0.350 (0.050) ($\prec$)
UF9	**0.388 (0.044)**	0.413 (0.059) ($\prec$)	**0.371 (0.035)**	0.400 (0.031) ($\prec$)
UF10	**1.145 (0.381)**	1.378 (0.292) ($\prec$)	**2.387 (0.510)**	3.135 (0.373) ($\prec$)
WFG1	**1.511 (0.056)**	1.649 (0.053) ($\prec$)	1.371 (0.057)	1.396 (0.059) ($\sim$)
WFG2	**0.303 (0.104)**	0.320 (0.104) ($\prec$)	0.343 (0.103)	0.346 (0.077) ($\sim$)
WFG3	0.181 (0.035)	0.193 (0.046) ($\sim$)	0.198 (0.021)	0.206 (0.028) ($\sim$)
WFG4	**0.083 (0.012)**	0.095 (0.023) ($\prec$)	0.160 (0.016)	0.164 (0.017) ($\sim$)
WFG5	**0.121 (0.011)**	0.149 (0.019) ($\prec$)	0.096 (0.008)	0.095 (0.009) ($\sim$)
WFG6	**0.194 (0.050)**	0.238 (0.076) ($\prec$)	0.183 (0.055)	0.188 (0.067) ($\sim$)
WFG7	0.146 (0.038)	0.132 (0.032) ($\sim$)	0.232 (0.039)	0.220 (0.024) ($\sim$)
WFG8	0.275 (0.029)	0.278 (0.033) ($\sim$)	0.357 (0.028)	0.363 (0.028) ($\sim$)

5 Conclusion

We addressed the problem of expensive optimization by proposing a machine learning based mechanism that aimed at improving the quality of the obtained Pareto front, under the assumption that the allowable computational budget is strictly limited. To speed up evolution, we formulated a regression learning task tailored to decomposition based EMO, in particular MOEA/D. The estimation of the meta-objective function was proposed to be optimized using the Support Vector Regression method. The knowledge was in turn incorporated into the evolution driven by MOEA/D through the use of an extended sampling step and a model-based filtering mechanism. We validated the proposed approach on various benchmark problems and the simulation results showed that the proposed learning enhanced EMO managed to obtain statistically significantly improved estimated Pareto front when the allowed computational budget is small.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Chia, J., Goh, C., Shim, V., Tan, K.: A data mining approach to evolutionary optimisation of noisy multi-objective problems. International Journal of Systems Science 43(7), 1217–1247 (2012)
3. Deb, K.: Multi-objective optimization. In: Multi-Objective Optimization using Evolutionary Algorithms, pp. 13–46 (2001)
4. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
5. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation 1(2), 61–70 (2011)
6. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)
7. Loshchilov, I., Schoenauer, M., Sebag, M.: A mono surrogate for multiobjective optimization. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 471–478. ACM (2010)
8. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation 12(5), 1207–1245 (2000)
9. Seah, C.W., Ong, Y.S., Tsang, I.W., Jiang, S.: Pareto rank learning in multi-objective evolutionary algorithms. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2012)
10. Tenne, Y., Goh, C.K.: Computational intelligence in expensive optimization problems, vol. 2. Springer (2010)
11. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
12. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms. Technical Report (2008)
13. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
14. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)

High against Low Quantile Comparison for Biomarker and Classifier Evaluation

Katharina Tschumitschew[1] and Frank Klawonn[2,3]

[1] ISC Gebhardt, Leonhardtstr. 19, D-29227 Celle, Germany
[2] Department of Computer Science, Ostfalia University of Applied Sciences
Salzdahlumer Str. 46/48, D-38302 Wolfenbuettel, Germany
[3] Bioinformatics and Statistics, Helmholtz Centre for Infection Research
Inhoffenstr. 7, D-38124 Braunschweig, Germany

Abstract. We propose a method to check whether a – usually high – quantile from a sample is smaller than another – usually low – quantile from another sample. It is only required that the two samples come from (independent) continuous, but not necessarily the same distributions. The proposed method is based on confidence intervals (CI) and can be used for the evaluation of classifiers for two class problems as an alternative to ROC analysis and in the context of biomarkers. The quantiles correspond to the acceptable false positive and false negative rates. We also introduce a visualisation that shows how the probabilities for the overlap of CIs we compute change when the chosen quantiles are varied. This can help to select a suitable combination of quantiles, i.e. the acceptable false positive and false negative rate.

1 Introduction

A biomarker for a disease is a measurement of a substance – often in blood, urine or saliva – that is a strong indicator for the considered disease. Prostate-specific antigen (PSA) is one example of a popular biomarker for prostate cancer (see for instance [1]). A high PSA level is an indicator for prostate cancer. A general problem with biomarkers is that they are like PSA not perfect, i.e. one cannot simply define a threshold such that all persons with a value below the threshold do not have the corresponding disease and all patients above suffer from the disease. Even if there is no such perfect threshold, a biomarker might still be of high benefit. If a perfect separation between healthy (D_- population) and non-healthy (with respect to the considered disease, D_+ population) persons is impossible, one must accept a certain fraction of misclassifications in the sense of false positives, healthy persons that are wrongly classified as having the disease, and false negatives, patients who are wrongly classified as healthy.

More generally speaking, we are faced with a classification problem with D_- and D_+ and we can use a measurement, the biomarker, to distinguish between the two classes. In classification, this measurement is usually called score. It does not even have to be a specific measurement. It is also possible that this score is derived from a classifier like logistic regression, naive Bayes, a neural network or a support vector machine. This classifier might not only rely on a single biomarker, but on a combination of measurements. So we will only use the term score in the following, referring either to

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 561–568, 2013.

the value of a biomarker or a value coming already from a classifier. We need to define a threshold that best separates the D_- from the D_+ population. The scores of the two classes follow two different distributions. Choosing a specific threshold means that we accept a certain percentage of false positives, represented by the right tail of the D_- distribution above the threshold, and a certain percentage of false negatives represented by the left tail of the D_+ distribution below the threshold. Since we usually do not know the distributions, we can only estimate them based on the samples. But even if we have good estimations of these distributions, we still have to decide about the suitable threshold yielding a certain trade-off between false positives and false negatives.

Receiver operating characteristic (ROC) curves (see for instance [2,3]) including their extension with confidence bands [4] are also used to deal with this situation. A ROC curve plots the true positive rate vs. the false positive rate of a classifier, depending on the threshold. However, ROC curves themselves provide only point estimates for the trade-off between true and false positives and the CIs tend to be very large.

Here we consider the fraction of misclassifications, i.e. the fractions of false positives α_{fp} and of false negatives α_{fn}. If we assume that the D_+ and D_- examples are sampled from two random variables X and Y with different unknown distributions, we would require that the $(1-\alpha_{\text{fp}})$-quantile of the D_- distribution is smaller than the α_{fn}-quantile of the D_+ distribution. Therefore, we are interested in comparing two different quantiles from two different random variables whose distributions we do not know.

In Section 2 we briefly review related work on the comparison of quantiles. Then, in Section 3, we provide a test-like approach to check the α-quantile of a random variable is larger than the β-quantile of another one. This approach allows us to compute a measure similar to p-values. One can plot these values depending on the choice of the quantiles to see how well the classes can be separated (see Section 4).

2 Related Work

Only a few approaches treat the problem of comparing arbitrary quantiles for two independent samples. The main focus of the test presented in [5] is the comparison of quantiles for double-censored data or data with repeated measures. Only equal quantiles are compared which is not what we need. The test provided in [6] and its extension in [7] are based on two independent CIs for the quantile of interest corresponding to the two populations. However, again only equal quantiles are compared.

3 Confidence Interval Overlap Probabilities

Let X and Y be independent continuous random variables with unknown cumulative distribution functions F_X and F_Y, respectively. Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ be i.i.d. random samples from X and Y. We are interested in the following question: $q_p^X < q_g^Y$, i.e. is the p-quantile of X smaller than the g-quantile of Y? The estimation of the population quantile q_p is given by the corresponding sample quantile $\hat{q}_p$. Several methods exist for the computation of $\hat{q}_p$, however roughly by all of them the p-th sample quantile is given by computing the real value $h = p \cdot n$. The value h is either rounded to

an integer or an interpolation between the two order statistics $X_{\lfloor h \rfloor}$ and $X_{\lceil h \rceil}$ nearest to the h-value is used. As in [6], we use CIs in order to compare two population quantiles. Since ties or discrete data cause specific problems for CI computation [8], we consider only continuous random variables in this work.

An obvious choice for the CI bounds is to choose two order statistics with the following property:

$$\mathrm{P}\left(X_{(l)} < q_p < X_{(r)}\right) = 1 - \alpha. \tag{1}$$

Here $1 - \alpha$ represents the confidence level. Since the probabilities $\mathrm{P}\left(X_{(l)} < q_p\right)$ and $\mathrm{P}\left(q_p < X_{(r)}\right)$ are calculated by binomial distributions, i.e. discrete distributions, the exact value $1 - \alpha$ can in general not be achieved. Therefore, instead of the equality in Eq. (1) we use "greater than or equal to" (see [9]). Therefore we must find two indices l and r, such that $1 \leq l < r \leq n$ and the inequality $\mathrm{P}\left(X_{(l)} < q_p < X_{(r)}\right) \geq 1 - \alpha$ holds. The CI (1) is illustrated in Figure 1.

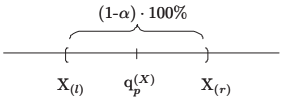


Fig. 1. $(1 - \alpha) \cdot 100\%$ CI for population quantile q_p

Given order statistics $X_{(1)}, \ldots, X_{(n)}$, the $(1 - \alpha) \cdot 100\%$ CI for the p-th quantile is given by Eq. (2) (see for instance [9,8]).

$$\mathrm{P}\left(X_{(l)} < q_p < X_{(r)}\right) = \sum_{i=l}^{r-1} \binom{n}{i} p^i \left(1-p\right)^{n-i} \geq 1 - \alpha. \tag{2}$$

The indices l and r are not unique, a set of combinations fulfills Eq. (2). l and r could be found separately by dividing the probability α between two tails. Therefore, l and r are chosen in such a way that the difference $(r - l)$ is minimal for given α, i.e. the biggest l and smallest r which satisfy

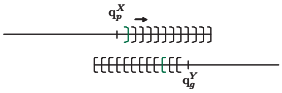
$$\sum_{i=0}^{l-1} \binom{n}{i} p^i \left(1-p\right)^{n-i} \leq \frac{\alpha}{2} \quad \text{and} \quad \sum_{i=0}^{r-1} \binom{n}{i} p^i \left(1-p\right)^{n-i} \geq 1 - \frac{\alpha}{2}. \tag{3}$$

In some cases, there may be no bounds $l - 1$ or $r - 1$ satisfying (3). This depends on the choice of n, p and α and occurs mainly when a CI for an extreme quantile should be computed. A too small sample size or a too high confidence level could also be a reason. The computation of the CI presented above is a distribution free method, i.e. we do not make any assumptions – except continuity – about the distributions. Since we are interested in the question whether $q_p^X < q_g^Y$ holds, it is sufficient to compute one-sided CIs. In case of X a right-sided and in case of Y a left-sided CI. Therefore we need to find indices $r^{(X)}$ and $l^{(Y)}$ for one-sided CI. Using Eq. (3), we obtain

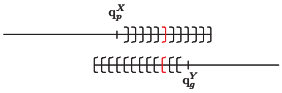
$$\sum_{i=0}^{r^{(X)}-1} \binom{n}{i} p^i \left(1-p\right)^{n-i} \geq 1 - \alpha_X \quad \text{and} \quad \sum_{i=0}^{l^{(Y)}-1} \binom{m}{i} g^i \left(1-g\right)^{m-i} \leq \alpha_Y. \tag{4}$$

We choose the α's in such a way that the CIs do not intersect, i.e. $X_{\left(r^{(X)}\right)} < Y_{\left(l^{(Y)}\right)}$. The probability for at least one of the quantiles to be outside of the CI is given by the risk value[1] r-value $= 1 - (1 - \alpha_X)(1 - \alpha_Y)$. We use the term r-value to distinguish it from a p-value used in hypothesis testing. We want to minimise the r-value over all possible pairs α_X and α_Y. For this purpose we use a brute force search strategy. Since the search space is restricted to order statistics, the complexity for brute force search is given by the product of the number of values right of q_p^X and left of q_g^Y. The basic idea of the algorithm can be described as follows.

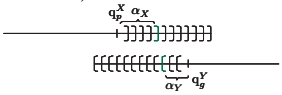
- Compute all possible CI bounds for q_p^X and q_g^Y.
- Choose those one-sided CIs for q_p^X and q_g^Y, such that the confidence bounds $X_{\left(r^{(X)}\right)}$ and $Y_{\left(l^{(Y)}\right)}$ do not overlap. Among these choose the ones that yield the smallest r-value. The search can be carried out in the following way.
  - For each CI bound of q_g^Y start with the closest bound for q_p^X and check whether the corresponding CIs intersect.



  - If they do not intersect, take the next bound for q_p^X.
  - Stop when the intervals start to intersect.



  - Compute the r-value for the last non-intersecting intervals. This yields the r-value $= 1 - (1 - \alpha_X)(1 - \alpha_Y)$.



  - Select among all computed r-values the smallest one.

4 Visualisation and Experimental Results

In this section we present an experimental evaluation of our proposed approach and a visualisation technique that can be applied when the quantiles cannot be specified exactly. We consider upper quantiles of X and lower quantiles for Y for different distributions.

- Normal distributions: $X \sim \mathrm{N}(0, 1)$ and $Y \sim \mathrm{N}(4.5, 1.5)$
- Exponential distributions with parameter λ: $X \sim \mathrm{Exp}(4)$ ($\lambda = 4$) and $Y \sim \mathrm{Exp}(0.1)$ ($\lambda = 0.1$)
- Exponential and normal distributions: $X \sim \mathrm{Exp}(4)$ and $Y \sim \mathrm{N}(3, 1)$

For all described examples we restrict our experimental evaluation to the quantiles from the following intervals: $q_X \in [80\%,\ 99.9\%]$ and $q_y \in [0.1\%,\ 20\%]$. The samples have equal size and contain 1000 observations ($m = n = 1000$). The results for the two

[1] The r-value is neither a p- nor a q-value used in hypothesis testing.

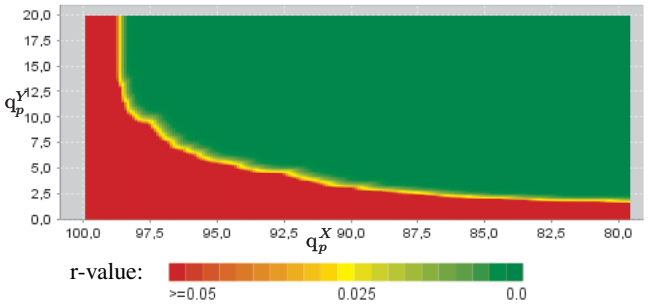


Fig. 2. r-Values for samples from two normal distributions: $X \sim \mathrm{N}(0, 1)$ and $Y \sim \mathrm{N}(4.5, 1.5)$

normal distributions ($\mathrm{N}(0, 1)$ and $\mathrm{N}(4.5, 1.5)$) are illustrated in Fig. 2. The x-axis represents the quantile q_p^X in decreasing order and the y-axis the quantile q_g^Y in increasing order. The r-values for the chosen quantiles q_p^X and q_g^Y are computed on a discrete grid for the plot in Fig. 2. In the following examples we use the discrete grid with a sampling rate of 0.1%. The colours in the plot characterise the r-values, all values less or equal to the selected significance level α are painted red, (perfect) green corresponds to the value zero. For other r-values ($0 <$ r-value $< \alpha$) the colours correspond to interpolation between red and green in such a way that for the r-value $= \frac{\alpha}{2}$ the corresponding colour is yellow. Hence we would expect the left lower corner to be painted with red, the right upper corner with green and gradient colours in between. In this example, the significance level α was chosen to be 0.05. Therefore, the orange/yellow path divides the plot into two subareas: the region where the r-value is too large and where it is small enough. In such a way we can better concentrate on the quantiles which might be interesting.

Of course, the computed r-values are also stored in a table. In Fig. 2, choosing the 90%-quantile for X and the 2.8%-quantile for Y leads to a sufficiently small r-value $= 0.0412$, which is smaller than $\alpha = 0.05$. For $q_{0.95}^X$ the first Y-quantile for which the r-value is smaller than $\alpha = 0.05$ is $q_{0.052}^Y$. The r-value is in this case equal to 0.4713.

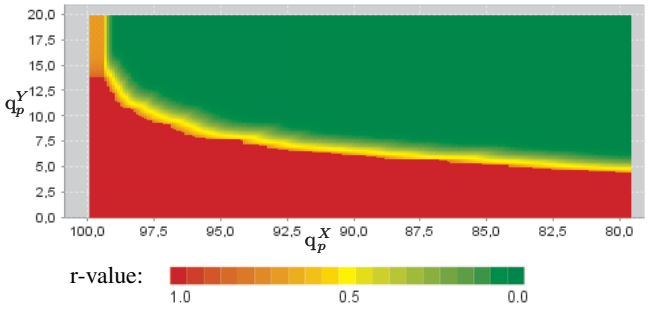


Fig. 3. r-Values for samples from two exponential distributions, $X \sim \mathrm{Exp}(4)$ and $Y \sim \mathrm{Exp}(0.1)$

The results for samples from two exponential distributions are presented in Fig. 3. It should be noted that we use a different r-value scale in this figure, red corresponds to the r-value $= 1$, the yellow colour represents r-value $= 0.5$. This kind of visualisation can be important in case of a first rough analysis of the data. In such a way we can easily choose which region is especially interesting for further examination. For $q^X_{0.90}$ the r-value becomes smaller than 1 for the quantile $q^Y_{0.058}$ for a sample of size 1000, for a sample size 10000 for $q^Y_{0.056}$. A more detailed analysis of the effects of sample size on the r-values is given at the end of this section.

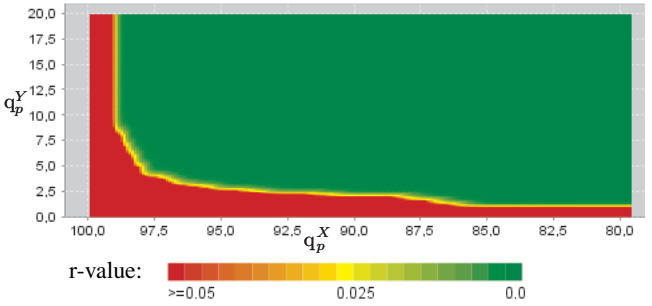


Fig. 4. r-Values for samples from an exponential and a normal distribution: $X \sim \text{Exp}(4)$ and $Y \sim \text{N}(3, 1)$

Except for the continuity, our approach does not make assumptions about the distributions of the two samples. X and Y may follow distributions from different families. Fig. 4 illustrates such an example. Here $X \sim \text{Exp}(4)$ and $Y \sim \text{N}(3, 1)$. The 90% X-quantile is approximately equal to the 0.767% Y-quantile. As shown in the plot, the r-values are smaller than 0.05 for example for the pair of quantiles $q^X_{0.90}$, $q^Y_{0.02}$, i.e. 90% and 2%. For the 99.5% and the 8.7% quantiles, the r-value is also smaller than 0.5, whereas for the 99.6% X-quantile it is not for Y-quantiles from the interval $[0.1\%,\ 20\%]$. In fact $q^X_{0.996} \approx q^Y_{0.053}$. The cause of such misleading large r-values is that an extreme X-quantile has to be tested and therefore the sample contains not enough observations for that purpose. So for instance by increasing the sample size of X to 10000, the corresponding r-value is small enough for the 9% Y-quantile.

As mentioned above, reliable estimations of the confidence bounds become increasingly difficult for extreme quantiles. One reasons is that for extreme quantiles there might be not enough values left over on the tails. Then the r-values will not be small enough although they should be theoretically. Then larger sample sizes are needed Fig. 5 shows an example of such a situation. Here $X \sim U(0, 1)$ and $Y \sim U(0.8, 1.8)$. The r-values were computed for q_X and q_Y quantiles ($q_X \in [80\%,\ 99.9\%]$ and $q_y \in [0.1\%,\ 20\%]$) and samples of different sizes. Fig. 5(a) shows the r-values for the sample size $n = 10$ (for each of the two samples). It can be seen that the methods yields only r-values larger than 0.5. A slightly different situation is presented in Fig. 5(b) for the two samples each containing $n = 20$ observations. The upper left corner is green, which means that the r-values are smaller than 0.05. For the sample size of $n = 50$

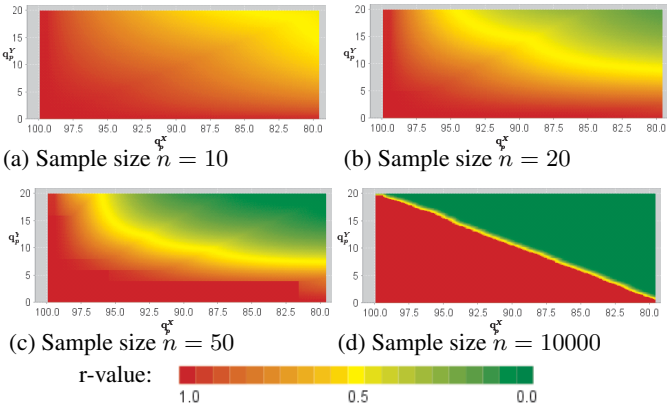


Fig. 5. Effects of the small sample sizes on the trustworthiness of the r-value

the situation improves even more (see Fig. 5(c)). The best r-values are obtain in Fig. 5(d) where a samples size of $n = 10000$ was chosen. This corresponds more or less to the theoretical values for the two given uniform distributions. Of course, in real world applications such large sample sizes will be seldom available. But we should keep in mind that we need to have a sufficient amount of data in order to get significant results.

In order to clarify the exact development of the r-values, we analysed the result of the test for two fixed quantiles. Fig. 6 shows the r-value curve for $q_X = 0.9$ and $q_Y = 0.15$ for different sample sizes. To avoid random fluctuations, for each sample size the mean r-value over 1000 repeated runs was computed. The x-axis represents the different sample sizes and the y-axis the corresponding r-value. As can be seen from Fig. 6, the first time the r-value becomes less than 0.05 is for two samples containing approximately 850 observations.

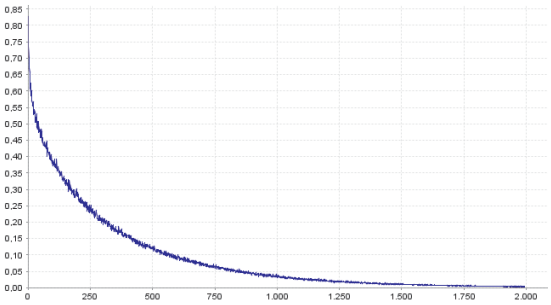


Fig. 6. r-values for $q_X = 0.9$ and $q_Y = 0.15$ for different sample sizes.

According to the considerations above it would be pointless to perform the test for quantiles very close to the extreme values close to 100% and 0%.

5 Conclusions

We have presented an approach that can check whether a – usually high – quantile from a sample is smaller than another – usually low – quantile from another sample. It is only required that the two samples come from (independent) continuous distributions. The two samples can come from different distributions. The proposed method is useful in the context of the evaluation of classifiers for two class problems as an alternative to ROC analysis and in the context of biomarkers.

We have also developed a visualisation technique that frees the user from specifying specific quantiles in advance. For very small sample sizes, the r-values will usually not be significant. But the visualisation can also be used to estimate the required sample sizes based on a small sample. Assuming that the general distributions of the two samples do not change, one can simply apply a bootstrapping technique by replicating the samples. However, a very small amount of noise must be added because we assume that the samples come from continuous distributions and ties in the samples would cause problems.

Acknowledgement. The research leading to these results has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement n° 115523-2 – COMBACTE, resources of which are composed of financial contribution from the European Union's Seventh Framework Programme (FP7/2007-2013) and EFPIA companies in kind contribution.

References

1. De Angelis, G., Rittenhouse, H., Mikolajczyk, S., Blair, S., Semjonow, A.: Twenty years of PSA: From prostate antigen to tumor marker. Reviews in Urology 9(3), 113–123 (2007)
2. Obuchowsky, N., Lieber, M., Wians Jr., F.: ROC curves in clinical chemistry: Uses, misuses and possible solutions. Clinical Chemistry 50, 1118–1125 (2004)
3. Søreide, K.: Receiver-operating characteristic (ROC) curve analysis in diagnostic, prognostic and predictive biomarker research. Clinical Pathology 62, 1–5 (2009)
4. Macskassy, S., Provost, F.: Confidence bands for ROC curves: Methods and an empirical study. In: Hernández-Orallo, J., Ferri, C., Lachiche, N., Flach, P. (eds.) ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI 2004, pp. 61–70 (2004)
5. Kosorok, M.: Two-sample quantile tests under general conditions. Biometrika 86(4), 909–921 (1999)
6. Ozturk, O., Balakrishnan, N.: Exact two-sample nonparametric test for quantile difference between two populations based on ranked set samples. Annals of the Institute of Statistical Mathematics 61(1), 235–249 (2009)
7. Wolfe, D.: Ranked set sampling: an approach to more efficient data collection. Statistical Science 19(4), 636–643 (2004)
8. David, H., Nagaraja, H.: Order statistics. John Wiley, New York (2003)
9. Gibbons, J., Chakraborti, S.: Nonparametric statistical inference. Chapman and Hall/CRC, Boca Raton (2010)

Learning a Label-Noise Robust Logistic Regression: Analysis and Experiments

Jakramate Bootkrajang and Ata Kabán

School of Computer Science, University of Birmingham,
Edgbaston, Birmingham, B15 2TT, UK
{J.Bootkrajang,A.Kaban}@cs.bham.ac.uk

Abstract. Label-noise robust logistic regression (rLR) is an extension of logistic regression that includes a model of random mislabelling. This paper attempts a theoretical analysis of rLR. By decomposing and interpreting the gradient of the likelihood objective of rLR as employed in gradient ascent optimisation, we get insights into the ability of the rLR learning algorithm to counteract the negative effect of mislabelling as a result of an intrinsic re-weighting mechanism. We also give an upper-bound on the error of rLR using Rademacher complexities.

Keywords: label noise, logistic regression, robust learning, gradient ascent optimisation, generalisation error bounds.

1 Introduction

In the context of supervised learning, a classification rule is to be derived from a set of labelled examples. The training sample is assumed to be drawn i.i.d. from some unknown distribution over tuples of the form $(\mathbf{x}, y)$, where $\mathbf{x}$ is an m-dimensional vector that represents a data point in the m-dimensional space and y is its class label assignment. In this paper we will consider linear classifiers. Geometrically, the classification rule is defined by a hyperplane that separates the classes. Regardless of the learning approach used, in the traditional framework of supervised learning, the induction of the classification rule crucially relies on the given class labels. Unfortunately, often in practice there is no guarantee that the class labels are all correct.

Noise in the labels may originate from the subjective nature of the labelling task, noisy communication channels, or lack of information to determine the true label of a given instance, to name just a few of the most common causes. The presence of class label noise in training samples has been empirically reported to deteriorate the performance of the existing classifiers in a broad range of classification problems including biomedical data analysis [7,13] and image classification [10,17]. Although, the problem posed by the presence of class label noise is acknowledged, often it gets naively ignored in practice.

There is an increasing research literature that aims to address the issues related to learning from samples with noisy class labels. [4,5,15,14]. Most previous

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 569–576, 2013.

approaches try to detect mislabelled instances based on various heuristics, and very few take a principled modelling approach such as either a generative [10] or a discriminative model [2,12]. The latter is our focus in this paper. While there is theoretical analysis on the negative effects of label noise for traditional logistic regression [8,9,6,11], an analysis of the good performance of a label-noise robust logistic regression approach has been lacking, and this is what we address here.

2 Robust Logistic Regression

Consider a set of training data $S = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^m$ represent an observation and $\tilde{y}_i \in \{0, 1\}$, denotes the given label of $\mathbf{x}_i$. In the classical scenario for binary classification, the log likelihood for logistic regression model is defined as:

$$L(\mathbf{w}) = \sum_{i=1}^{n} \tilde{y}_i \log(\tilde{P}_i^1) + (1 - \tilde{y}_i) \log(\tilde{P}_i^0), \tag{1}$$

where $\tilde{P}_i^k := p(\tilde{y}_i = k|\mathbf{x}_i, \mathbf{w})$. If the labels were presumed to be correct, we would have

$$p(\tilde{y} = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{(-\mathbf{w}^T\mathbf{x})}}, \tag{2}$$

and whenever this is above 0.5 we would decide that $\mathbf{x}$ belongs to class 1. Here, $\mathbf{w}$ is the weight vector orthogonal to the decision boundary and it determines the orientation of the separating plane. However, when there is label noise present, then it is no longer valid to make predictions in this way. Instead we introduce a latent variable y, to represent the true label, and we rewrite $\tilde{P}_i^k$ as the following:

$$\tilde{P}_i^k = \sum_{j=0}^{1} p(\tilde{y}_i = k|y_i = j)p(y_i = j|\mathbf{x}_i, \mathbf{w}) \tag{3}$$

where, $\gamma_{jk} := p(\tilde{y}_i = k|y_i = j)$ represents the probability that the label has flipped from the true label j to the observed label k. It is worth noting that γ_{jk} is a global value for all observations. These additional parameters form a transition table Γ of class label flippings and these parameters need to be estimated.

The flipping probabilities for a two-class problem are summarised in Table 2. The table will be referred to as the 'gamma matrix' from now on. Hence,

Table 1. Probabilistic relationships between given label and true label

		$\tilde{y}$	
		0	1
y	0	γ_{00}	γ_{01}
	1	γ_{10}	γ_{11}

instead of predicting the label using the posterior probability of $\tilde{y}$ as in eq. (2), we are now able to calculate the posterior probability of the true label y, and use this to decide that $\mathbf{x}$ belongs to class 1 whenever $p(y = 1|\mathbf{x}, \mathbf{w}) \geq 0.5$.

3 Results

3.1 Understanding rLR Learning Via an Interpretation of Gradient Ascent Optimisation

We have seen in the last section that rLR makes use of a latent variable y to model the true label. However, it is not obvious at all why the introduction of this extra variable will lead to a more robust model. In this section we give an intuitive explanation for why this is the case indeed, by looking at the gradient ascent optimisation process for learning the rLR model. The gradient update rule for the j-th iteration is the following:

$$\mathbf{w}^j = \mathbf{w}^{j-1} - \eta \times \mathbf{g} \tag{4}$$

where η is usually referred to as the 'learning rate' and $\mathbf{g} = \nabla_{\mathbf{w}} L(\mathbf{w})$ is the gradient of the likelihood objective function.

Assume a fixed training set S of size n, and assume also that the gamma matrix is known. Let n_1 be the number of points that have been assigned the label $\tilde{y} = 1$ and n_0 the number of points with $\tilde{y} = 0$. Denote by $\mathbf{g}^+$ the terms of the gradient that correspond to points with $\tilde{y} = 1$, and $\mathbf{g}^-$ those of points with $\tilde{y} = 0$, so that $\mathbf{g} = \mathbf{g}^+ + \mathbf{g}^-$. For classical logistic regression without label noise modelling, these terms are:

$$\begin{aligned}
\mathbf{g}^+ &= \frac{\partial \sum_{i=1}^{n} \tilde{y}_i \log(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}_i}})}{\partial \mathbf{w}} &= \sum_{i=1}^{n_1} [p(y_i = 0|\mathbf{x}_i, \mathbf{w}) \cdot \mathbf{x}_i] \\
\mathbf{g}^- &= \frac{\partial \sum_{i=1}^{n} (1-\tilde{y}_i) \log(\frac{e^{-\mathbf{w}^T\mathbf{x}_i}}{1+e^{-\mathbf{w}^T\mathbf{x}_i}})}{\partial \mathbf{w}} &= \sum_{i=1}^{n_0} [-p(y_i = 1|\mathbf{x}_i, \mathbf{w}) \cdot \mathbf{x}_i]
\end{aligned} \tag{5}$$

Let us now look at the corresponding two terms of the gradient for our robust logistic regression. From the definition of data likelihood using eqs. (1)-(3) we get:

$$\mathbf{g}^+_{rLR} = \sum_{i=1}^{n_1} \left[\frac{\tilde{y}_i(\gamma_{11} - \gamma_{01})}{\tilde{P}_i^1} \sigma(\mathbf{w}^T\mathbf{x}_i)(1 - \sigma(w^T\mathbf{x}_i)) \cdot \mathbf{x}_i \right]$$

Now, define $\alpha_i = \frac{(\gamma_{11}-\gamma_{01})p(y_i=1|\mathbf{x}_i,\mathbf{w})}{\tilde{P}_i^1}$, and we can rewrite this as:

$$\mathbf{g}^+_{rLR} = \sum_{i=1}^{n_1} [\alpha_i \cdot p(y_i = 0|\mathbf{x}_i, \mathbf{w}) \cdot \mathbf{x}_i] \tag{6}$$

Likewise, define $\beta_i = \frac{(\gamma_{00}-\gamma_{10})p(y_i=0|\mathbf{x}_i,\mathbf{w})}{\tilde{P}_i^0}$, and find:

$$\mathbf{g}^-_{rLR} = \sum_{i=1}^{n_0} [-\beta_i \cdot p(y_i = 1|\mathbf{x}_i, \mathbf{w}) \cdot \mathbf{x}_i] \tag{7}$$

We see that both α_i and β_i act as weighting coefficients. This weighting mechanism adjusts the contribution that particular data points make to the gradient. The weight itself, for example α_i is composed of two components, the

first is a global weight: $(\gamma_{11} - \gamma_{01})$; the second is a data point specific weight: $p(y_i = 1|\mathbf{x}_i, \mathbf{w})/\tilde{P}_i^1$. That is, each data point within a class will be weighted equally by the global weight, in accordance with the extent of label noise in that class. In addition, the individual points that have been potentially mislabelled are multiplied by the data-point-specific weighting factor.

Thus, $\mathbf{g}_{rLR}^+$ will take into account the wrong information to a lesser extent. Similar reasoning applies to $\mathbf{g}_{rLR}^-$in a symmetric manner. Through this weighting mechanism, rLR is then able to distinguish between correctly labelled points and mislabelled points. As a consequence, it is expected that rLR will perform better in terms of generalisation error in label noise conditions, and will be able to detect mislabelled instances with high accuracy.

Finally, we find it instructive to look at the behaviour of robust logistic regression in a label-noise free scenario, i.e. when the training labels are actually all correct. In this case $\tilde{y} = y$, $\gamma_{01} = 0$ and $\gamma_{11} = 1$. In consequence $\alpha_i = 1$, and likewise $\beta_i = 1$. Hence is this case we recover $\mathbf{g}_{rLR}^+ = \mathbf{g}^+$ and $\mathbf{g}_{rLR}^- = \mathbf{g}^-$ as in classical logistic regression.

3.2 Error Analysis

Although there is no framework to analyse the error of predicting the true labels since those are hidden even in the training phase, in this section we derive a bound on generalisation error in predicting $\tilde{y}$ for a new input $\mathbf{x}$. Such bound is informative because it gives a guarantee that the rLR model (that includes the hidden variable y) is able to explain the *observed* data pairs.

We will use Rademacher complexities. Let $l(h, \mathbf{x}, \tilde{y})$ denote the loss of a classifier (or hypothesis) $h \in \mathcal{H}$ on the input point $\mathbf{x}$, where $\mathcal{H}$ is the hypothesis class considered. Define $L_D(h) = \mathbb{E}_{(\mathbf{x},\tilde{y})\sim D} l(h, \mathbf{x}, \tilde{y})$ to be the generalisation error, and $L_S(h) = (1/n) \cdot \sum_{i=1}^n l(h, \mathbf{x}_i, \tilde{y}_i)$ to be its empirical estimate. The Rademacher complexity of the composite function of the hypothesis class $\mathcal{H}$ and a training set S is defined as:

$$R(\mathcal{H} \circ S) = \frac{1}{n} \mathop{\mathbb{E}}_{\sigma\sim\{\pm 1\}^n} \left[\sup_{h\in\mathcal{H}} \sum_{i=1}^n \sigma_i h(\mathbf{x}_i) \right] \tag{8}$$

If we can obtain the Rademacher complexity for our classifier then we can use the following lemma, which gives an upper bound on the generalisation error of an Empirical Risk Minimisation (ERM) classifier in terms of its Rademacher complexity.

Lemma 1 (Barlett and Mendelson [1]). *For a training set S of size n, let h_s be an ERM hypothesis. Assume that for all $\mathbf{x} \in S$ and $h \in \mathcal{H}$ we have the ρ-Lipschitz loss function l satisfying $|l(h, \mathbf{x}, \tilde{y})| \leq c$ for some positive constant c. Then, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$*

$$L_D(h_S) - L_S(h_S) \leq 2\rho R(\mathcal{H} \circ S) + c\sqrt{\frac{\log(1/\delta)}{2n}} \tag{9}$$

To apply this lemma, we first define the loss function $l(h, \mathbf{x}, \tilde{y})$ associated with rLR that we will show to be Lipschitz. We then calculate the Rademacher complexity of the hypothesis class of rLR. Finally, the generalisation error bound is obtained by plugging the Rademacher complexity into Lemma 1.

Let us begin with the loss function associated with robust logistic regression. Recall the data log-likelihood in eq. (1). Because of monotonicity of logarithm, maximising the log-likelihood is equivalent to minimising the negative log-likelihood. It then follows that we can define the loss function to be the negative of the data log-likelihood:

$$l(h, \mathbf{x}_i, \tilde{y}_i) = -\tilde{y}_i \log(\tilde{P}_i^1) - (1 - \tilde{y}_i) \log(\tilde{P}_i^0) \tag{10}$$

We are now ready to consider the Rademacher complexity of rLR. It has been shown that the complexity of the hypothesis class of a linear halfspace classifier is bounded as:

Lemma 2 (Adapted from Shalev-Shwartz [16]). *Let $\mathcal{H} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle, \|\mathbf{w}\|_2 \leq \lambda\}$ defines the hypothesis class of linear classifiers. Let $S = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of vectors in $\mathbb{R}^m$. Then,*

$$R(\mathcal{H} \circ S) \leq \frac{\lambda \max_i \|\mathbf{x}_i\|_2}{\sqrt{n}} \tag{11}$$

Note that rLR can be regarded as a linear classifier because the loss is a function of a linear function of $\mathbf{w}$. Hence the complexity of its hypothesis class is also defined by Lemma 2. Using these, we state and prove our result as the following:

Theorem 1. *Let $h_s \in \mathcal{H}_{rLR}$ be an ERM hypothesis from the class of rLR classifiers, and let l be the loss function defined in eq. (10). Let S be a training set of n examples drawn i.i.d. from an unknown distribution over $\mathbb{R}^m$. We assume that $\|\mathbf{x}_i\|_2 < \infty, \forall i = 1 : n$, and also that γ_{00} and γ_{11} are bounded away from zero. Then, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$ the following bound holds:*

$$L_D(h_s) \leq L_S(h_s) + 2\frac{\lambda \max_i \|\mathbf{x}_i\|_2}{\sqrt{n}} + c\sqrt{\frac{\log(1/\delta)}{2n}} \tag{12}$$

Proof. We first show that the rLR loss function is a Lipschitz function with Lipschitz constant $\rho = 1$. Note that $\tilde{y}$ can either take value 0 or 1 but not both at the same time, so the loss function decouples into two terms. In order to obtain the Lipschitz constant, we shall show that the absolute value of the derivative of each term is bounded by 1. Without loss of generality, let us consider the first term, $\log(\tilde{P}_i^1)$. Also for convenience, we define $\mathbf{a} = \mathbf{w}^T \mathbf{x}$. Then,

$$\begin{aligned}
\tilde{P}_i^1 &= \frac{\gamma_{11}}{1 + e^{-\mathbf{a}}} + \frac{\gamma_{01} e^{-\mathbf{a}}}{1 + e^{-\mathbf{a}}} = \frac{\gamma_{11} + \gamma_{01} e^{-\mathbf{a}}}{1 + e^{-\mathbf{a}}} \\
\frac{\partial \log(\tilde{P}_n^1)}{\partial \mathbf{a}} &= \frac{\partial \log(\gamma_{11} + \gamma_{01} e^{-\mathbf{a}}) - log(1 + e^{-\mathbf{a}})}{\partial \mathbf{a}} = \left| \frac{-\gamma_{01} e^{-\mathbf{a}}}{\gamma_{11} + \gamma_{01} e^{-\mathbf{a}}} + \frac{e^{-\mathbf{a}}}{1 + e^{-\mathbf{a}}} \right| \\
&= \left| \frac{1}{1 + e^{\mathbf{a}}} - \frac{1}{1 + \frac{\gamma_{11}}{\gamma_{01}} e^{\mathbf{a}}} \right| = |f(\alpha)|,
\end{aligned} \tag{13}$$

where in the last step we divided through by $e^{-\mathbf{a}}$, and we defined $\alpha := \gamma_{11}/\gamma_{01}$. Since γ_{11} and γ_{01} are probabilities, their values are between 0 and 1. It follows that the domain of α is $[0, \infty)$. Observe that eq. (13) attains its maximum either 1) at its end points $f(0)$ and at the limit $f(\infty)$ or 2) at a point where $f'(\alpha) = 0$. The first case can be easily checked by plugging the extreme values into eq. (13), and we see that $f(0) \in [-1, 0]$ and $f(\infty) \in [0, 1]$. The remaining case can be verified by calculating $f'(\alpha), \alpha \in (0, \infty)$ which turns out to be,

$$f'(\alpha) = \frac{e^{\mathbf{a}}}{(1 + \alpha e^{\mathbf{a}})^2} \tag{14}$$

It can be seen that eq. (14) is non-negative and can only be zero when $\alpha = \infty$, which is the case we have already considered. Hence, we learn that the absolute value of eq. (13) is bounded by 1. It follows that $\log(\tilde{P}_i^1)$ is a 1-Lipschitz function. $\log(\tilde{P}_i^0)$ can be shown to be 1-Lipschitz using similar technique.

Next we need to show that the loss function is bounded. Without loss of generality, let us consider eq. (10) where $\tilde{y}_i = 1$. It can be shown that the term is bounded by some finite number c.

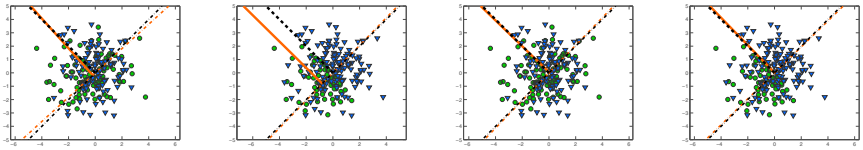
$$\begin{aligned}
\left| -\log(\tilde{P}_i^1) \right| &= \left| -\log\left(\frac{\gamma_{01} e^{-\mathbf{a}}}{1 + e^{-\mathbf{a}}} + \frac{\gamma_{11}}{1 + e^{-\mathbf{a}}}\right) \right| \\
&= \left| \log(1 + e^{-\mathbf{a}}) - \log(\gamma_{01} + \gamma_{11} e^{-\mathbf{a}}) \right| \\
&\leq \left| \log(1 + e^{-\mathbf{a}}) \right| + \left| \log(\gamma_{01} + \gamma_{11} e^{-\mathbf{a}}) \right| \\
&\leq 1 + ||\mathbf{w}||_2 ||\mathbf{x}||_2 + \left| \log(\gamma_{01} + \gamma_{11} e^{-\mathbf{a}}) \right|
\end{aligned} \tag{15}$$

where we have used the triangle inequality to get the third line. The last line makes use of a bound on the loss function of standard logistic regression which takes the form $|\log(1 + e^{-\mathbf{a}})| \leq 1 + ||\mathbf{w}||_2 ||\mathbf{x}||_2$. For the second term we will show that its value is finite, that is we show that an argument to logarithm is never be 0 or infinity. Since $\mathbf{a} = \mathbf{w}^T \mathbf{x}$ is bounded because of the assumptions and, γ_{11} is bounded away from zero, it can be shown that 1) the maximum value of $\gamma_{01} + \gamma_{11} e^{-\mathbf{a}}$ is finite, and 2) its minimum is bounded away from 0. As a consequence, we have that the loss function of rLR is bounded by some positive constant c, as required.

Finally, since we have that the loss function of robust logistic regression is 1-Lipschitz, and it is bounded by a finite constant c, we apply lemma 1 to conclude the statement of our theorem. □

4 Experiments

We first illustrate the effect of two types of label noise, symmetric and asymmetric using synthetic data. The decision boundary obtained from both models are plotted in Fig. (1). It is obvious that symmetric label noise does not alter the decision boundary of traditional logistic regression as expected. By contrast, LR suffers from asymmetric mislabelling. Robust logistic regression, however, was able to perform well in both situations. Next we demonstrate the benefit of having label noise modelling using four UCI data sets: *Adult*, *Boston*, *Liver*, and



(a) LR-symm. (b) LR-asym. (c) rLR-symm. (d) rLR-asym.

Fig. 1. Decision boundary obtained by LR and rLR at 40% symmetric and 40% asymmetric label noise respectively. The orange lines are the estimates and the black lines are the true values.

Pima. More applications of rLR and L1-regularised rLR may be found in our previous works [2] and [3]. We also compare rLR with its generative counterpart: robust Normal Discriminant Analysis (rNDA) [10]. We first show the result on clean datasets where the labels are all perfect. This is to demonstrate that label noise formulation does not deteriorate the performance when there is no wrong label. The mean error and standard deviation from 50 independent runs are presented in Table 2 (top). It is clear from the result that the rLR reduces to classical LR when there is no wrong label.

Further, we artificially inject 10% and 30% asymmetric label noise into the training sets. We hold out 20% of the data for testing and train on the remaining points. Table (2) (middle and bottom) summarises the empirical results. As expected, and in agreement with our analysis, rLR improves upon traditional LR in label noise conditions. Observe also that rNDA is competitive only when the Gaussianity assumption holds true, and is inferior to rLR, otherwise.

Table 2. Empirical error rate of LR vs rLR vs rNDA in predicting the true labels from partially mislabelled training set. The mislabelling rate was 10% and 30%, of asymmetric type. P-values from Wilcoxon test at 5% level are presented. Boldface highlights statistically best method(s) while italics indicates the second best method.

Dataset	Error at 0 % Noise			p-values		
	rNDA	LR	rLR	rNDA v. LR	rNDA v. rLR	LR v. rLR
Adult	22.70 ± 0.03	**19.87 ± 0.02**	**20.13 ± 0.02**	$6.26e^{-6}$	$4.81e^{-5}$	0.62
Boston	15.53 ± 0.03	**13.86 ± 0.03**	**13.08 ± 0.03**	0.022	0.001	0.23
Liver	33.74 ± 0.05	32.03 ± 0.05	31.33 ± 0.04	0.075	0.02	0.39
Pima	24.84 ± 0.03	23.47 ± 0.03	24.36 ± 0.04	0.073	0.50	0.26
	10 % Noise					
Adult	22.06 ± 0.04	**19.94 ± 0.03**	**19.70 ± 0.02**	0.001	$6.52e^{-4}$	0.84
Boston	16.62 ± 0.04	*14.55 ± 0.03*	**13.06 ± 0.03**	0.015	$1.26e^{-5}$	0.021
Liver	33.80 ± 0.06	32.14 ± 0.06	32.20 ± 0.06	0.15	0.092	0.81
Pima	23.53 ± 0.03	22.77 ± 0.03	23.66 ± 0.03	0.34	0.72	0.14
	30 % Noise					
Adult	*24.88 ± 0.07*	26.87 ± 0.05	**21.66 ± 0.05**	0.023	0.008	$4.17e^{-7}$
Boston	*20.27 ± 0.04*	22.29 ± 0.05	**14.94 ± 0.04**	0.044	$3.31e^{-8}$	$1.19e^{-10}$
Liver	37.68 ± 0.09	39.36 ± 0.07	**33.62 ± 0.07**	0.11	0.032	$2.11e^{-4}$
Pima	**24.35 ± 0.04**	28.05 ± 0.04	**25.09 ± 0.04**	$6.28e^{-6}$	0.30	$4.22e^{-4}$

5 Conclusion and Future Work

We have presented a theoretical analysis of the label-noise robust logistic regression (rLR) classifier, showing that its error is bounded, and that rLR behaves the same as LR when there is no label noise or when the label flipping is symmetric. We have also demonstrated that rLR improves upon LR when there is asymmetric label flipping, and this is due to a weighting mechanism that our analysis has revealed. This was achieved by decomposing and interpreting the gradient as employed in gradient ascent optimisation. Future work is required to modify the formalism in a way to have a loss function defined on the latent true label rather than the observed noisy label.

References

1. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: risk bounds and structural results. JMLR 3, 463–482 (2003)
2. Bootkrajang, J., Kabán, A.: Label-noise robust logistic regression and its applications. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part I. LNCS, vol. 7523, pp. 143–158. Springer, Heidelberg (2012)
3. Bootkrajang, J., Kabán, A.: Classification of mislabelled microarrays using robust sparse logistic regression. Bioinformatics 29(7), 870–877 (2013)
4. Brodley, C.E., Friedl, M.A.: Identifying and eliminating mislabeled training instances. In: Proceedings of AAAI 1996, pp. 799–805 (1996)
5. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167 (1999)
6. Chhikara, R.S., McKeon, J.: Linear discriminant analysis with misallocation in training samples. Journal of the American Stat. Assoc. 79(388), 899–906 (1984)
7. Krishnan, T., Nandy, S.C.: Efficiency of discriminant analysis when initial samples are classified stochastically. Pattern Recognition 23(5), 529–537 (1990)
8. Lachenbruch, P.A.: Discriminant analysis when the initial samples are misclassified. Technometrics 8(4), 657–662 (1966)
9. Lachenbruch, P.A.: Discriminant analysis when the initial samples are misclassified ii: Non-random misclassification models. Technometrics 16(3), 419–424 (1974)
10. Lawrence, N.D., Schölkopf, B.: Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of ICML 2001, pp. 306–313 (2001)
11. Lugosi, G.: Learning with an unreliable teacher. Pattern Recogn. 25, 79–87 (1992)
12. Magder, L.S., Hughes, J.P.: Logistic regression when the outcome is measured with uncertainty. American Journal of Epidemiology 146(2), 195–203 (1997)
13. Malossini, A., Blanzieri, E., Ng, R.T.: Detecting potential labeling errors in microarrays by data perturbation. Bioinformatics 22(17), 2114–2121 (2006)
14. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. Journal of Intelligent Information Systems 22(1), 89–109 (2004)
15. Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters 24(7), 1015–1022 (2003)
16. Shalev-Shwartz, S.: Introduction to machine learning. The Hebrew University of Jerusalem (2009), `http://www.cs.huji.ac.il/~shais/Handouts.pdf`
17. Yasui, Y., Pepe, M., Hsu, L., Adam, B.-L., Feng, Z.: Partially supervised learning using an EM-boosting algorithm. Biometrics 60(1), 199–206 (2004)

Hybrid Bacterial Foraging Algorithm for Data Clustering

Ben Niu[1,2,3], Qiqi Duan[1], and Jing Liang[4,*]

[1] College of Management, Shenzhen University,
Shenzhen 518060, China
[2] Department of Industrial and System Engineering,
Hong Kong Polytechnic University, Hong Kong
[3] Hefei Institute of Intelligent Machines, Chinese Academy of Sciences,
Hefei 230031, China
[4] School of Electrical Engineering, Zhengzhou University, Zhengzhou, China
liangjing@zzu.edu.cn

Abstract. Data clustering is a fundamental but challenging task in many research fields, such as pattern recognition, data statistics and machine learning. Various clustering techniques such as *K-means* have been proposed for compact clustering solutions. However, these clustering methods are sensitive to the selection of initial cluster centers and suffer from the problem of premature convergence. Recently, novel swarm intelligence algorithms such as bacterial foraging algorithm (BFA) offer some inspirations for the design of clustering algorithms, owing to their promising global and parallel search capacities. In this paper, we propose a hybrid clustering algorithm (BFCA) based on BFA and *K-means*. The proposed algorithm attempts to take full advantages of global search capacities of BFA and excellent local search abilities of *K-means*. Experimental results on four datasets show that the proposed technique outperforms *K-means* in terms of clustering quality.

Keywords: Data clustering, *K-means* clustering technique, bacterial foraging algorithm (BFA), benchmark datasets.

1 Introduction

Data clustering is a fundamental task for researchers from diverse disciplines, such as data mining, pattern recognition, knowledge discovery, data statistics, artificial intelligence, and machine learning [1]. Clustering is a process of partitioning a given collection of data objects of any size and sample into a number of non-overlapping and meaningful clusters based on some similarity criteria [2]. However, data clustering is a challenging task, as the increasing amount of high-dimensional data makes clustering harder without prior knowledge. Generally, clustering problems, which can be translated into combinational optimization problems, are considered as one type of NP-hard problems [3].

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 577–584, 2013.

To solve the problem, a variety of clustering algorithms have been proposed, which aim to minimize specified evaluation functions (or called fitness functions) such as similarity/dissimilarity measurements. One of the most commonly used clustering algorithms is *K-means*, which has been successfully applied to a wide range of clustering problems. However, related studies have shown that *K-means* is prone to falling into local optimal convergence and is sensitive to the configuration of initial cluster centroids [1]-[4].

Essentially, the process of searching for the optimal cluster centroids in feature space can correspond to discovering the optimal positions with the best fitness in the multi-dimensional search space. Recently, swarm intelligence algorithms have been developed for optimization and clustering, due to their promising global and parallel search capacities [5]-[11]. For example, Murthy and Chowdhury [8] applied genetic algorithms (GAs) to clustering problems. GAs obtained the representative clustering compared with *K-means* on four benchmark datasets. A similar approach can be found in Ref. [9]. The difference lies in that Krishna [9] hybridized GAs with the distance-based mutation and gradient descent steps of *K-means*. Besides, Handl and Meyer [10] identified two types of ant-based clustering techniques and summarized their corresponding merits and drawbacks. Further, Nouaouria et al [11] reviewed a variety of clustering approaches based on particle swarm optimization and provided some insights for swarm-based clustering.

However, to the best of the authors' knowledge, little substantial work can be conducted to apply the bacterial foraging algorithm (BFA) to clustering problems. The original bacterial foraging algorithm (BFA) mimics how the E. coli bacteria obtain nutrients [12]. Though faced with various geological or physiological constraints, bacteria perform complex behaviors to maximize nutrients obtained. The bacterial foraging processes, which are comprised of chemotaxis, reproduction, dispersal and elimination, are regarded as the distributed optimization processes. For more details of BFA, please refer to Ref. [12]. The goal of this paper is to investigate whether BFA can be applied to data clustering by means of hybridizations with *K-means*.

The rest of this paper is organized as follows. Section 2 illustrates in particular the proposed algorithm, followed by experimental studies and discussions in Section 3. Finally, Section 4 concludes the research and further work.

2 Bacterial Foraging Clustering Algorithm

In this section, we proposed a bacterial foraging clustering algorithm (BFCA), which combines both the merits of BFA and *K-means*. In nature, the bacteria exhibit two basic movement patterns (or called chemotaxis). It tumbles in a random direction, or swims in a direction of increasing nutrients for a period of time [12]. As the chemotaxis operations provide the driving force for global search, we mainly take into account the combination between chemotaxis and *K-means*.

After many generations, the bacteria with sufficient nutrients possess the high opportunity of breeding. In opposition, the bacteria with poor foraging strategies are

more likely to be killed. Owing to the selective breeding, the bacteria can maintain the sustainable development of colonies. Besides, the number of bacteria may fluctuate due to environment changes or emergency events. Although the dispersal-elimination events disturb the chemotactic process, they may be beneficial to keep the diversity of population and jump the local optimum [12].

K-means is viewed as one of center-based iterative clustering techniques, where each cluster is characterized by a center (or centroid), usually an average of all the data points within a cluster [1]-[7]. It often chooses the Euclidean distance to measure the similarity within a cluster. If a data point has the shortest Euclidean distance to some centroid, it would belong to the cluster represented by that centroid. The objective of *K-means* as well as the proposed algorithm is to minimize the sum of the intra-cluster Euclidean distances. The pseudo-code of BFCA is described below:

1. **Initialize** the population P randomly, and each individual i represents K cluster centroids where $P_i=(C_1, C_2, \ldots, C_k)$.
2. **Dispersal-Elimination loop** (l=1 to N_{ed}):
3. **Reproduction loop** (r=1 to N_{re}):
4. **Chemotaxis loop** (j=1 to N_c):
5. **Tumbling**: all the bacteria move in a random direction D where C represent the run length and *Delta* is a random number.

$$D_i(j+1,r,l) = D_i(j,r,l) + C * Delta \tag{1}$$

6. **Swimming**: the bacteria which find better food move in the favorable direction until N_s steps are reached.
7. Conduct ***K-means*** Operations for all the bacteria.

A. Assign all the data objects i($i=1,2,\ldots,N$) to the closest cluster C_j $(j=1,2,\ldots,K)$ according to similarity measurements.

B. Repeat the following operations for each data object x_i:

C. Calculate the centroid (Z_i) using Equation (2) where n_j is the number of data objects within the j-th cluster (Note that the centroids are adjusted both **during and after** each partitioning).

$$Z_i = \sum_{i \in C_j} x_i \Big/ n_j ; (i = 1,2,..,N; j = 1,2,...,K) \tag{2}$$

D. Calculate the Euclidean distances (D_{ij}) between the i-th data object and all centroid using Equation (3);

$$D_{ij} = \left\| x_i - z_j \right\|^2 \tag{3}$$

E. If *i*-th data object located in the *j*-th cluster has shorter Euclidean distances to another cluster *m* (Note that *m* is not equal to *j*), it changes the membership and moves to the *m*-th cluster. If *i*<*N*, go to Step (A); otherwise, go to step (F).
F. Calculate the sum of the within-cluster Euclidean distances using Equation (4).

$$E = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| x_{ij} - Z_j \right\|^2 \tag{4}$$

G. Stop until the stopping criteria are satisfied (when the maximum of iterations has been reached, or when there are few changes in the cluster membership).
8. **End** the Chemotaxis loop when $j=N_c$.
9. Conduct reproduction operations. One half of bacteria enjoy the chance of propagation by replacing another half.
10. **End** the Reproduction loop when $r=N_{re}$.
11. Conduct the dispersal-elimination events. Each bacteria is mitigated another area with a specified probability.
12. **End** the dispersal-elimination loop when $l=N_{ed}$.

For the purpose of study, the stopping criterion in this paper is the predefined maximum number of iterations.

3 Experiments and Discussions

To measure and confirm the performance of the bacterial foraging clustering algorithm (BFCA), we compare it with *K-means*.

3.1 Experimental Setup and Datasets Chosen

For a fair comparison, all algorithms are executed on 20 independent runs under the MATLAB environment (Matlab7.10.0R2011a) in the same machine. For each algorithm, the maximal number of iterations is set as 10000. In addition, we choose two artificial datasets [13] shown in Fig.1 and Fig.2, and two real-world datasets derived from the well-known UCI Machine Learning Repository available at [14]. The artificial dataset 1 is comprised of 4 groups, each of which has 400 data samples with 4 attributes, and the artificial dataset 2 includes 800 data objects and 4 features, which can be gathered into 4 clusters. The Wine dataset has 13 attributes and 178 samples divided into 3 clustering. The Iris dataset consists of 3 clusters where each cluster has 50 data samples with 3 features.

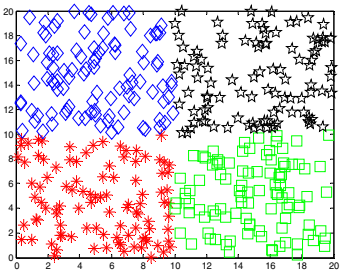


Fig. 1. Artificial Dataset 1

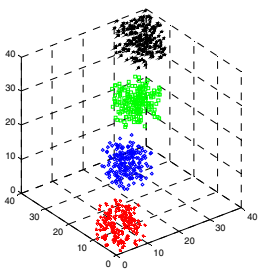


Fig. 2. Artificial Dataset 2

After trial and error, the parameter settings of BFCA are presented as follows. The number of chemotaxis, reproduction, and elimination-dispersal events is set to 10, 5, and 2, respectively. The run length and swimming length is equal to 2 and 4, and the number of bacteria is set as 100. The averages and standard deviation of error rate, sum of within-cluster distances and execution time obtained are recorded, as de-scribed in Table 1. The number of clusters is predefined for clustering algorithms involved here.

3.2 Comparison Results

As we can see from Table 1, BFCA yields more compact clustering results than *K-means* for all the datasets in terms of error rate and sum of within-cluster distances. However, *K-means* manages to outperform its opponent in terms of execution time in all the considered cases.

Table 1. Results Obtained by BFCA and *K-means*

Datasets		Error Rate (%)	Within-cluster distances	Execute Time (seconds)
Artificial Dataset1	*K-means*	1.61(1.74e-3)	2.75e+2(7.45e+2)	**1.08e+0(2.67e-4)**
	BFCA	**0.55(6.05e-4)**	**2.08e+2(1.54e+2)**	4.45e+2(7.31e+1)
Artificial Dataset2	*K-means*	0.63(7.97e-4)	3.43e+2(2.24e+3)	**2.26e+0(2.01e-3)**
	BFCA	**0.31(3.08e-4)**	**2.21e+2(1.68e+3)**	1.05e+3(8.32e+3)
Wine Dataset	*K-means*	4.81(2.50e-3)	4.53e+2(1.66e+2)	**4.98e-1(5.21e-5)**
	BFCA	**3.26(5.32e-6)**	**3.89e+2(1.05e+2)**	2.03e+2(3.66e+1)
Iris Dataset	*K-means*	16.41(2.97e-4)	1.28e+2(2.36e-2)	**4.11e-1(1.03e-4)**
	BFCA	**15.71(1.82e-5)**	**1.19e+2(2.71e-2)**	1.46e+2(1.88e+02)

It is can be observed in Fig.3 and Fig.4 that *K-means* can obtain faster convergence in the early stage of iterations (about below 50 iterations) for all the datasets. However, *K-means* quickly gets stuck in the local optima in the starting stage. On the contrary, BFCA can escape from the local optima from approximately 50 iterations. It may be due to the fact that the parallel scheme adopted by the proposed algorithm enhanced its global search abilities.

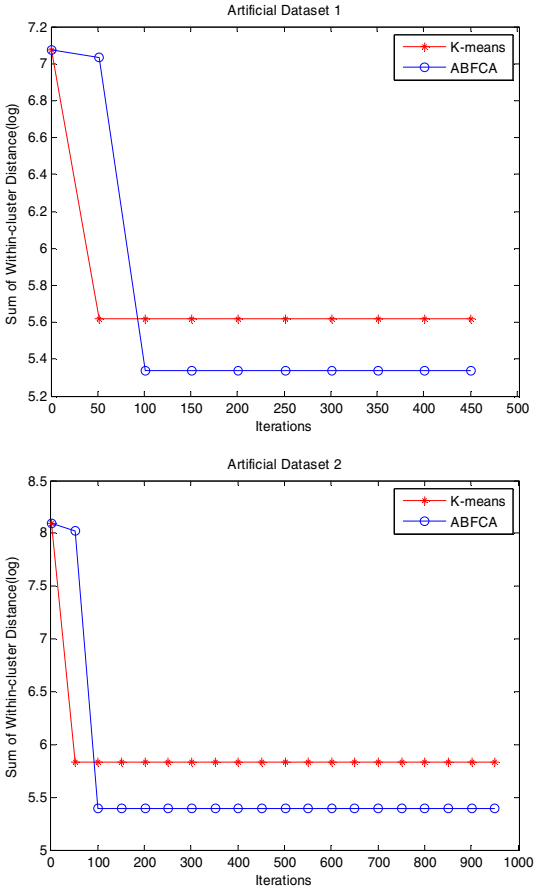


Fig. 3. Convergence of Sum of Within-cluster Distances for Dataset 1 and 2 Obtained by BFCA and *K-means*

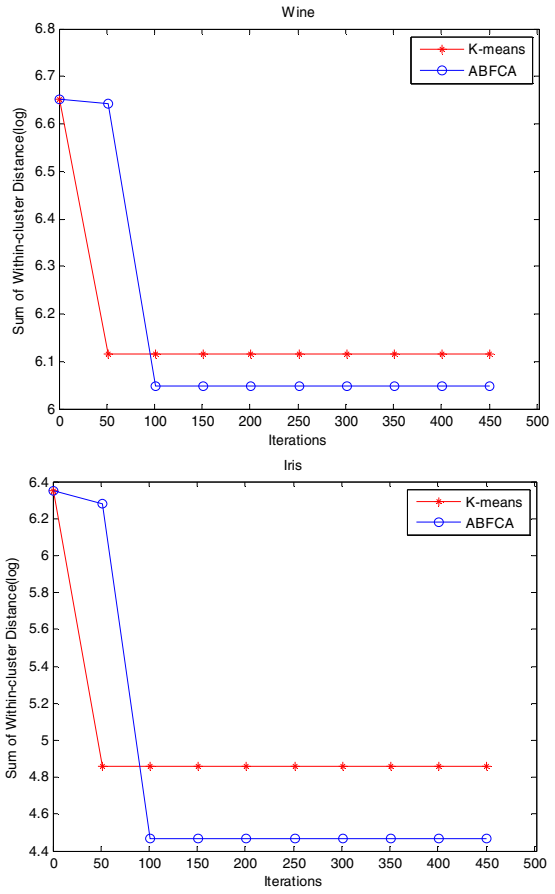


Fig. 4. Convergence of Sum of Within-cluster Distances for Wine and Iris Obtained by BFCA and *K-means*

4 Conclusions and Further Work

In this paper, we propose a hybrid clustering algorithm called BFCA based on the bacterial foraging algorithm and *K-means*. Experimental results obtained have been presented, showing that BFCA is able to provide the satisfactory performances on four datasets selected. Although we cannot claim that the proposed algorithm can perform well on all datasets, we can see its potential applications on data clustering. Besides, since BFCA is computationally expensive, ongoing research is aimed at improving its implementations in term of computation speed.

Acknowledgments. This work is partially supported by The National Natural Science Foundation of China (Grants nos. 71001072, 71271140, 60905039), The Hong Kong Scholars Program 2012 (Grant no. G-YZ24), China Postdoctoral Science Foundation

(Grant nos. 20100480705, 2012T50584), Science and Technology Project of Shenzhen (Grant No. JC201005280492A) and the Natural Science Foundation of Guangdong Province (Grant nos. S2012010008668, 9451806001002294).

References

1. Flynn, P.J., Murty, M.N., Jain, A.K.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
2. Likasa, A., Vlassisb, N., Verbeekb, J.: The Global K-means Clustering Algorithm. Pattern Recognition 36(2), 451–461 (2003)
3. Jain, A.K.: Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
4. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Grouping Multidimensional Data, pp. 25–71 (2006)
5. Nouaouria, N., Boukadoum, M., Proulx, M.: Particle Swarm Classification: A Survey and Positioning. Pattern Recognition 46(7), 2028–2044 (2013)
6. Martens, D., Baesens, B., Fawcett, T.: Editorial Survey: Swarm Intelligence for Data Mining. Machine Learning 82(1), 1–42 (2001)
7. Das, S., Abraham, A., Konar, A.: Automatic Clustering Using an Improved Differential Evolution Algorithm. IEEE Transactions on System, Man, and Cybernetics–Part A: Systems and Human 38(1), 218–237 (2008)
8. Murthy, C.A., Chowdhury, N.: In Search of Optimal Clusters Using Genetic Algorithms. Pattern Recognition Letters 17(8), 825–832 (1996)
9. Krishna, K., Murty, M.N.A.: Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics 29(3), 433–439 (2009)
10. Handl, J., Meyer, B.: Ant-based and Swarm-based Clustering. Swarm Intelligence 1(2), 95–113 (2007)
11. Nouaouria, N., Boukadoum, M., Proulx, R.: Particle Swarm Classification: A Survey and Positioning. Pattern Recognition 46(7), 2028–2044 (2013)
12. Passion, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine, 52–67 (2002)
13. Kao, Y.T., Zahara, E., Kao, I.W.: A Hybridized Approach to Data Clustering. Expert Systems with Applications 34(8), 1754–1762 (2008)
14. UCI Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Swarm Intelligence with Clustering for Solving SAT

Habiba Drias, Ameur Douib, and Célia Hirèche

Computer Science Department,
LRIA, USTHB
Algiers, Algeria
hdrias@usthsb.dz

Abstract. Swarm intelligence is a major research field that contributed these last years to solve complex problems. In this paper, we show that bio-inspired approaches augmented with data mining techniques such as clustering, may bring more efficiency to problem solving. In fact, we aim at exploring judiciously the search space before seeking for solutions and hence reducing the complexity of the problem large instances. We consider for this purpose the approach of Bee Swarm Optimization (BSO) and propose two ways to integrate clustering in it. The first one consists in incorporating clustering in the design of BSO. This leads us to suggest an advanced version of BSO. The second one performs clustering on the data before launching BSO. This proposal was implemented for the satisfiability problem known widely as SAT. The complexity of the problem is reduced in this case by clustering clauses and hence variables and afterwards solving the clusters that have a smaller number of variables.

Keywords: Bee Swarm Optimization BSO, Clustering, satisfiability problem.

1 Introduction

Swarm intelligence has been developed over the past years for solving NP-complete problems among other orientations. Unlike the exact methods that guarantee to find the optimal solution if it exists but they are not able to tackle problems of large instances because they would generate a combinatorial explosion and a timeout calculation, bio-inspired approaches are based on approximation and do not guarantee to find a solution even if one exists. In order to reach more closely to optimal solutions, the search space should be explored in a judicious manner. If we consider the solutions collection as a set of data, data mining techniques may help acquiring some knowledge on the search space and then exploiting the achieved information in solving the problem. We are especially interested in locating those regions that are promising in the sense that they contain quality solutions. To go further with the idea, the data mining techniques that can meet this concern are clustering tasks.

Two ways of designing the whole approach are investigated. We consider Bee Swarm Optimization (BSO) to implement these features and the well known satisfiability problem (SAT) as an application. A way of reducing the complexity of SAT is to cluster the clauses into sub-problems with a smaller number of variables and maintaining a

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 585–593, 2013.

number of common variables between sub-problems as small as possible. Of course, we are in the context of problem reduction but the approach is very original.

In section 2, BSO is introduced and its general aspects are presented. The problem SAT is described in section 3 and the proposal and its implementation are discussed in section 4. The conducted experiments and the obtained results are exhibited in the last section.

2 Bee Swarm Optimization

Bee Swarm Optimization (BSO) was designed as a nature and bio-inspired approach to solve complex problems and was published the first time in 2005 [6]. Afterwards, it showed its efficiency through its adaptation to satisfiability, information retrieval [7] and association rule mining [8].

BSO was designed to translate as accurately as possible the behavior of bees when searching for food. Its framework is based on the experience held in 1991 by Seely, and Sneyd Camazine, who showed after experience that when two food sources exist equidistant from the hive, the bees exploit those whose concentration is highest. The bee exploiting the source of greater concentration makes a dance, which is proportional in strength to the source concentration.

A bee called BeeInit, represents a potential solution to the problem called Sref. A search space, namely SearchArea is then determined from Sref using a well-defined diversification generator. Each bee considers one solution from SearchArea as the starting point from which it performs a local search. After the bees have accomplished their search, they communicate their best results through a table called *Dance*. The best solution stored in this table is taken as the reference solution for the next iteration. In addition, a tabu list for backup references solutions *Sref* at each iteration is used to avoid stagnation and local optimum. The framework of BSO is outlined in Algorithm 1.

```
Algorithm 1. BSO
begin
1. Sref ← BeeInit, a random solution;
2. while stop condition is not reached do
 2.1. insert Sref in TL the Tabu List;
 2.2. determine SearchArea from Sref;
 2.3. assign for each bee a solution of SearchArea;
 2.4. for each bee do
     2.4.1. search in the corresponding area;
     2.4.2. insert result in Dance;
 2.5. Choose the best solution from Dance for Sref;
end;
```

3 The Satisfiability Problem

Satisfiability arouses more interest from the community of artificial intelligence and the computational complexity. Indeed, all eyes turn to this problem in recent years thinking that effective problem solving SAT could make significant progress in these

two major disciplines that have a significant impact in the field of computer science in general.

3.1 Definitions

Given a set of Boolean variables V = {v1, v2, ..., vn}, and an assignment function t: V → {T, F} that assigns a truth value to each variable vi then:

- A literal is a variable that appears with or without the negation operator.
- A clause is a disjunction of literals.
- An instance of SAT is a conjunction of clauses.
- The problem SAT is defined by the following (instance, question) pair
    - Instance: m clauses over n literals
    - Question: is there an instanciation of variables for which all the clauses are true?

A clause is True (T) if and only if there is at least one positive literal with the value T or a negative literal with the value False (F).

Example:
Let consider the following set of variables V = {v1, v2 v3, v4} and the following set of clauses C = {C1, C2, C3} defined as:
C1 = v1, v2, -v4
C2 = v2, -v3, v4,
C3 = -v1, -v2, v3
The '-' sign denotes the negation operator and ',' the disjunctive operator. One possible solution is the instanciation {T, T, T, T} respectively to {v1, v2 v3, v4}.

3.2 SAT Solvers

There are several approaches to solving SAT and a large number of SAT solvers have been developed since the 60s. They are used in the academic world as well as in the industry to solve complex problems. The first designed solvers are called "complete" because they are able to return the optimal solution if it exists or prove that there is none. The most known SAT solvers are:

The Davis-Putnam Procedure (DPP)
DPP [2] is one of the complete algorithm that was widely used for solving SAT. The idea of the algorithm is the construction of a binary tree using recursion and backtracking. This way the size of the search space is considerably reduced relatively to the methods that existed previously by eliminating variables from the instance. The process repeats selecting a variable v and assigning to it the possible Boolean values and then eliminating all clauses that contain v or –v.

The Davis-Putnam-Logemann-Loveland (DPLL)
DPLL [3] is an improved version of DPP w.r.t. the space complexity. Loveland and Logemann introduced the splitting rule, which recursively assigns Boolean values to a variable and solves the resulting sub-problems. When a clause is satisfied, it is suppressed and this operation is called simplification rule.

Other more modern and powerful SAT solvers based on DPLL [9,11] were developed and are more widely used these days. They are able to solve instances of hundreds of thousands of clauses over thousands of variables.

As we know, when the size of the instance to solve is very large, the exact algorithms can confront a combinatorial explosion during their execution and any high degree of performance of the actual machines cannot avoid this problem. So "incomplete" approaches were designed as an alternative to yield near-optimal solutions. Their main drawback is that they never guarantee to calculate the optimal solutions. There is nowadays a plethora of techniques, including incomplete heuristics and meta-heuristics. These methods are called incomplete because they explore only a part of the search space when the latter is prohibitive.

Several interesting incomplete SAT solvers were developed and they outperformed the complete solvers in some cases, among them we can cite:

GSAT [13] is a randomized local search. It starts by drawing randomly a valuation for the variables and then making a certain number of flips on variables that reduce the number of unsatisfiable clauses. This process is repeated until getting the optimal solution or reaching a limit on the number of tries.

Walksat [14] is an extended version of GSAT. A noise represented by a probability is introduced in the procedure to realize the random walk move. With a probability p a variable drawn at random is considered and with (1-p) the variable that yields the maximum satisfied clause is selected.

On the other hand, several meta-heuristics like Scatter search [4] and Tabu search [12] and bio-inspired approaches such as GA, ACO [5] and BSO [6] were adapted for SAT.

4 BSO with Clustering

We identified two ways to integrate clustering with BSO: an internal clustering and an external one. The internal clustering resides inside the framework of BSO whereas the external clustering is in a process that includes clustering and then BSO.

4.1 Advanced BSO

In traditional BSO, SearchArea is determined according to a diversification generator that produces locations in the search space that are equidistant from each other in order to browse a great part of it. It is based hence on pure mathematical computations of the distances that separates the different points of SearchArea. This way of doing things ignores completely the topology of the solutions space and do not take into account the potential locations of quality solutions. This results in searching in some unnecessary regions and lost of time. Advanced BSO is designed to improve its performance and instead of yielding SearchArea using a mathematical process, it will be generated by a clustering process. The new framework of BSO is described in Algorithm 2. The role of BeeInit is to determine SearchArea and hence here it is to cluster the clauses. Then each bee will be assigned a cluster, which is a subset of clauses to solve with an exact method such as DPLL.

```
Algorithm 2. Advanced BSO
begin
1. while stop condition is not reached do
   1.1. cluster the clauses; (* by BeeInit *)
   1.2. SearchArea ← the set of clusters centroids;
   1.3. assign for each bee an element of SearchArea;
   1.4. for each bee do
     1.4.1. solve the corresponding cluster;
     1.4.2. insert the unsatisfied clauses in Dance;
   1.5. clauses ← the clauses not yet satisfied;
end;
```

4.2 Clustering and BSO

The second framework is to create a process that includes a clustering followed by BSO, it is outlined in Algorithm 3.

```
Algorithm 3. Clustering and BSO
begin
1. cluster the clauses;
2. apply BSO on each cluster;
3. get the overall solution;
end;
```

4.3 An Optimal Clauses Clustering

The approach consists of two steps, the determination of cluster centers using a genetic algorithm and the clustering using k-means.

1) The Clusters Centers Determination

In order to select in an optimal way the clusters centers, we design a genetic algorithm where the individuals of the population represent the centers of potential clusters. Each individual is a set of clauses, where each of them is a center of a cluster. The genetic process called GA-CC follows the four steps of Algorithm 4.

```
Algorithm 4. GA-CC
1. generate randomly the individual centers.
2. apply Cross Over operators on the centers.
3. apply mutation on centers.
4. choose the best individuals of the resulting population.
```

For the last step, we need to consider a way to measure the distance between the clauses. We define this distance as the similarity between clauses in terms of common variables. The best selected centers should be the most distant as possible from each

other with a minimum of redundant variables between themselves. We use the following two functions to measure the individual quality:

- f1 calculates the distance between the centers.
- f2 counts the number of common variables between individuals (inter-center).

f1 must be maximized while f2 must be minimized. Hence, we take as a fitness function F, the following formula:

F = f1-f2

2) Clauses Clustering

We suggest a k-means algorithm to perform this task. Originally, this algorithm starts with random centers and we know that this way of generating initial centers weakens the algorithm performance. This is why the centers in our case are computed using a genetic algorithm. The combination of both algorithms optimizes the clauses clustering. Algorithm 5 outlines the framework of the overall clustering. The new center is computed as the means of all the clauses that are in the cluster. Concretely, the means for a cluster is equal to the median of all clauses contained in the cluster. The convergence is reached when no new centers are generated.

Algorithm 5. GA-k-means
1. generate the initial centers using algorithm 2.
2. repeat until convergence.
    2.1. insert each clause in the closest cluster.
    2.2. compute the new center for each cluster.

4.4 The Overall Solution

The resulting clusters are considered as new SAT sub-problems with a smaller number of clauses and hence less difficult to solve. Each cluster is treated separately and the set of clusters may also be solved in parallel. One issue arises and it resides in the fact that the clusters are not totally disjoints. This makes the resolution trickier in terms of building the overall solution and therefore the combination of solutions clusters taking into account the redundant variables.

Non-redundant variables take the value of the solution of the cluster in which they appear. Regarding common variables, assigning a value for a variable in a cluster and its opposite in another may arise a serious issue. All the remaining unsatisfied clauses will undergo another clustering processing.

Furthermore, an improvement technique was introduced: each time a cluster is solved, all its clauses that appear in the other clusters are removed. This operation reduces the complexity of solving the rest of the clusters.

5 Experimental Results

The experiments were conducted on all BMC (Bounded Model Checking) benchmarks [1] presented in Table 1.

Table 1. BMC benchmarks

benchmark	Benchmarks characteristics	
	Number of variables	*Number of clauses*
ibm 1	9685	55870
ibm 2	2 810	11683
ibm 3	14930	72106
ibm 4	28161	139716
ibm 5	9396	41207
ibm 6	51639	368352
ibm 7	8710	39774
galileo 8	58074	294821
galileo 9	63624	326999
ibm 10	59056	323700
ibm 11	32109	150027
ibm 12	39598	194778
ibm 13	13215	65728

These instances are hard to solve and used by the scientific community. Extensive experiments were undertaken in order to test the gain brought by the integration of data mining techniques in solving SAT with an incomplete approach such as BSO. The results are shown in Fig. 1. They show that the pre-clustering of clauses before starting the resolution yields much better performance in terms of satisfaction of clauses and in other terms that the gap existing between the performances of BSO and the improved approach is significant.

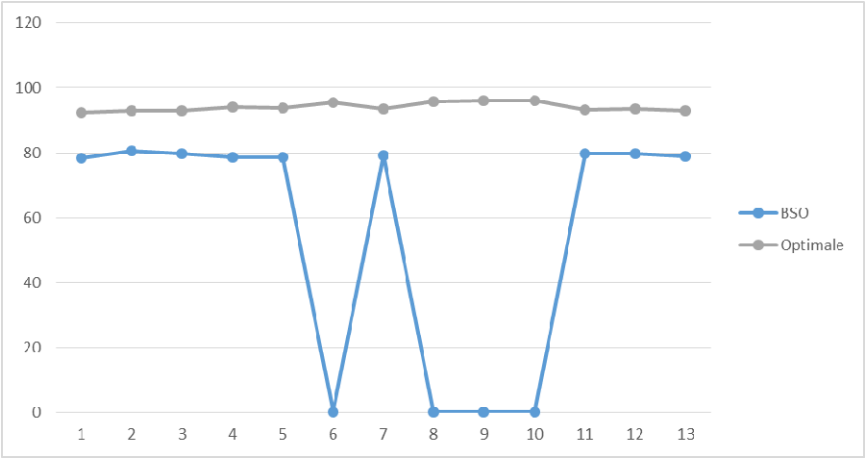


Fig. 1. Comparing BSO with Advanced BSO for SAT

Table 2 shows the execution time for Clustering+BSO and BSO. Clearly, we observe that not only Clustering+BSO is capable to solve all the instances but it is also faster.

Table 2. Running times for Clustering+BSO and BSO in seconds

benchmark	Clustering+BSO	BSO
ibm 1	27,74	677,65
ibm 2	5,63	57,15
ibm 3	49,64	809,17
ibm 4	188,78	4398,88
ibm 5	86,89	342,55
ibm 6	617,03	>>>
ibm 7	24,25	562,52
Galileo 8	1502,81	>>>
Galileo 9	1620,47	>>>
ibm 10	1038,02	>>>
ibm 11	229,63	6345,89
ibm 12	389,26	7652,18
ibm 13	39,18	716,96

6 Conclusion

Throughout this paper, we proposed an approach for solving problems by associating clustering techniques with a bio-inspired approach. The BSO clustering task aims at identifying promising regions for search in the solutions space. Two different manners of integrating clustering were proposed.

Experimental tests were performed on a collection of 13 BMC benchmarks to test satisfiability on these novel proposals. The results show the impact of clustering gave the best results in terms of satisfaction rate, but also in terms of processing time compared to the resolution using traditional BSO.

Furthermore, we believe that the use of algorithms such as "a priori" for association rules could give good results clustering, considering the terms as transactions and variables such items. Another perspective would be to parallelize the clusters resolution in order to gain in running time.

References

1. Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic Model Checking without BDDs. In: Cleaveland, W.R. (ed.) TACAS 1999. LNCS, vol. 1579, pp. 193–207. Springer, Heidelberg (1999)

2. Davis, M., Putnam, H.: A computing procedure for quantification theory. In: CACM, vol. 7, pp. 201–215 (1960)
3. Davis, M., Logemann, G., Loveland, D.: A Machine Program for Theorem Proving. Communications of the ACM 5(7), 394–397 (1962)
4. Drias, H., Khabzaoui, M.: Scatter search with random walk strategy for SAT and MAX-W-SAT problems. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 35–44. Springer, Heidelberg (2001)
5. Drias, H., Ibri, S.: Parallel ACS for weighted MAX-SAT. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 414–418. Springer, Heidelberg (2003)
6. Drias, H., Sadeg, S., Yahi, S.: Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 318–325. Springer, Heidelberg (2005)
7. Drias, H., Mosteghanemi, H.: Bees Swarm Optimization based Approach for Web Information Retrieval. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto Canada, pp. 6–13 (2010)
8. Djenouri, Y., Drias, H., Mosteghanemi, H.: Bees Swarm Optimization for Web Association Rule Mining. In: Problems and Methodologies in Mathematical Software Production, pp. 142–146. Springer LNCS (2012)
9. Gomes, C.P., Kautz, H., Sabharwal, A., Selman, B.: Satisfiability Solvers. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, pp. 89–134. Elsevier (2008)
10. Han, J., et al.: Data mining Concepts and Techniques. Morgan Kauffman Series (2013)
11. Le Berre, D., Roussel, O., Simon, L.: SAT competition (2007), http://www.satcompetition.org/
12. Mazure, B., Sais, L., Grégoire, E.: Tabu Search for SAT. In: AAAI/IAAI, pp. 281–285 (1997)
13. Selman, B., Levesque, H.J., Mitchell, D.G.: A new method for solving hard satisfiability problems. In: 10th AAAI, San Jose, CA, pp. 440–446 (1992)
14. Selman, B., Kautz, H., Cohen, B.: Local search strategies for satisfiability testing. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26, pp. 521–532. American Mathematical Society (1996)

Multilevel Bee Swarm Optimization for Large Satisfiability Problem Instances

Marwa Djeffal and Habiba Drias

Department of Computer Science
USTHB, LRIA, Algiers, Algeria
marwa.djeffal@gmail.com, hdrias@usthb.dz

Abstract. Nowadays, real word problems are often characterized by very large and complex search spaces. Finding high-quality solutions to NP-hard problems in a scalable and efficient manner has become a priority in the field of combinatorial optimization. This paper presents a new framework to handle large satisfiability problem instances (SAT) using a method designed by hybridizing two promising approaches. The first one is a bio-inspired metaheuristic called Bee Swarm Optimization (BSO) and the second one is the multilevel paradigm. The results obtained by comparing BSO with and without the multilevel concept showed that the latter improved significantly the performance of BSO.

Keywords: Large SAT problem instances, Scalability, swarm intelligence, multilevel paradigm, Bee Swarm Optimization.

1 Introduction

The fast technological advances have led to a significant growth rate of the size of the problems dealt with in different areas. With this new challenge known as scalability, the current trend is to create hybrid methods, striving to benefit of these approaches by combining them. In this paper, we propose a new framework based on the hybridization of two promising approaches: a bio-inspired metaheuristic called Bee Swarm Optimization [1], and the multilevel paradigm that has proven to be quite useful in handling large and very large graphs [2]. To prove its efficiency, we applied it on the satisfiability problem (SAT). This NP-Complete problem has become central in computational theory and artificial intelligence. Furthermore a lot of real-world problems can be naturally formulated as a SAT problem such as: automatic reasoning, planning, model checking, graph theory and VLSI.

The rest of the paper is organized as follows. In the next Section, we provide a description of the satisfiability problem as well as some known SAT solvers. Then, in Section 3, we present the Bee Swarm Optimization metaheuristic and in Section 4, we introduce the multilevel paradigm. Afterwards, we detail the phases of the proposed approach applied to the SAT problem in Section 5. In Section 6 we present the computation results of the conducted experiments. Finally, in Section 7 we conclude our work and give some interesting perspectives.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 594–602, 2013.

2 The Satisfiability Problem

2.1 Definition

SAT problem deals with propositional Boolean formulas. The standard encoding of a propositional Boolean formula is the Conjunctive Normal Form (CNF). A CNF formula F is a conjunction of clauses (c1,... cm), with each clause being a disjunction of literals (l1, ... ln), and each literal is an atomic Boolean variable xi or its negated form ¬xi. Assigning a Boolean variable xi to true satisfies all occurrences of the literal xi. A clause is satisfied if at least one of its literals is satisfied. Finally the CNF formula is satisfied if all its clauses are satisfied.

2.2 SAT Solvers

The algorithms for solving SAT fall into two main categories: systematic algorithms and metaheuristic algorithms. Systematic search algorithms are guaranteed to return a solution to a SAT instance if one exists or prove it unsatisfiable or inconsistent otherwise. The main systematic method for solving SAT problems is the DPLL [4]. It is a recursive and depth-first search-tree procedure, the worst-case complexity of solving SAT problems using the original DPLL is $O(m2^n)$ where m is the number of clauses and n is the number of variables. One way to overcome the problem of combinatorial explosion is to give up completeness. This led to consider metaheuristics as a mean to solve SAT problems. A detailed SAT history and methods of resolution, as GSAT, WalkSAT, evolutionary algorithms, can be found in [5].

One of the most recent works handling large SAT- encoded problems is [6]. It presents an interesting merging between a memetic algorithm (MA) and the multilevel paradigm. Apart from its wide uptake in the graph-partitioning area, multilevel refinement schemes are increasingly appearing for other combinatorial problems. The purpose then was to find out if the multilevel technique suites SAT problem solving techniques. The results showed that the multilevel technique boosted the performance of the MA.

3 Bee Swarm Optimization (BSO)

The nature-inspired metaheuristics that simulates the foraging behavior of real bee swarms became very popular lately [7].

```
Begin
let Sref be the solution found by BeeInit;
while (MaxIter not reached) do
    insert Sref in a taboo list;
    determine SearchArea from Sref;
    assign a solution of SearchArea to each bee;
    for each Bee k do
        search starting with the assigned solution;
        store the result in Dance;
    endfor;
    compute the new solution of reference Sref;
endwhile;
End.
```

The BSO metaheuristic proposed in [1], is based on a population of artificial bees cooperating together to solve an instance of an optimization problem. The general algorithm of BSO is as shown above. First, the bee guide named BeeInit searches for a solution presenting good features that will be called Sref, and from which the other solutions of the search space are computed via a certain strategy. This strategy flips a number of variables determined by the parameter Flip at certain positions of Sref so that the generated solutions will be different two by two. In fact, Flip represents the distance that separates the solutions between themselves and computed by the Hamming formula. This way, the strategy consists in exploring the whole search space by considering solutions that are dispersed. The set of these solutions is called SearchArea. Then, each of these solutions is assigned to a bee and becomes the starting point of the search in its region.

After accomplishing its search, every bee communicates through a table named Dance to its fellows the best solution visited. One of the solutions of this table will become the new solution of reference Sref for the next iteration of the process. In order to avoid cycles, Sref is stored every time in a taboo list. To ensure both good quality of the solution and good search space coverage, the selection of the reference solution at the end of an iteration of the search process follows the intensification and diversification principles. The algorithm stops when the optimal solution is found, or the maximum number of iterations is reached.

4 The Multilevel Paradigm

The multilevel concept has been used exclusively to solve graph partitioning problem (GPP) [2]. Thanks to the work of Walshaw, which has become recently a framework to enhance the performance of metaheuristics in different domains [8].The multilevel procedure is therefore outlined as follows:

input: problem P_0
begin
Level := 0
while (level < L) *//Coarsening Phase*
$P_{level+1}$:= Coarsen (P_{level})
Level := level+1
endwhile
S_L := Initial Solution (P_L) *//Initialisation Phase*
while (level > 0) *//Extension and refinement Phase*
S_{start} ($P_{level-1}$) := Extend (S_{final} , P_{level}) *//extend problem and project previous level's solution*
S_{final} ($P_{level-1}$):= Refine (S_{start} , $P_{level-1}$) *// refine the initial solution*
Level := level -1
endwhile
end.

As shown in the pseudo-code above the multilevel method is divided into three distinct phases:

Coarsening Phase: It is an iterative procedure that defines the hierarchical structure (P_0, P_1, ..., P_L) for a given problem $P = P_0$ where L is the number of levels. The coarsening phase, which is most likely the key component of a multilevel implementation, seems to hold three principles [2]. First, the coarsening should cease when any further coarsening would render the initialization degenerate. Second, a solution in any of the coarsened spaces should induce a legitimate solution on the original space. Thus, at any stage after initialization, the current solution could simply be extended through all the problem levels to achieve a solution of the original problem. Third, ideally, any solution in a coarsened space should have the same cost with respect to the objective function as its extension to the original space (i.e. the coarsening is exact): this requirement ensures that the coarsening is truly filtering the solution space. There is no general coarsening strategy, it depends mainly on the nature of the problem.

Initialization Phase: It produces an initial solution of P_L (the problem at the coarsest level). This could be done using a random assignment or a specific heuristic.

Extension and Refinement Phase: This phase is an iterated combination of two steps: The extension step where the solution found at the previous level is extended to give an initial solution to the problem at the current level. And the refinement step where a better solution is built from the initial one.

5 Proposed Multilevel BSO

Multilevel BSO (MLBSO) follows the multilevel scheme described earlier and uses classical BSO as a refinement algorithm. The choice of BSO resides in the simplicity of its fitness function combined with the global communication mode between the bees; therefore, MLBSO has four basic components:

5.1 Coarsening Algorithm

This algorithm coarsens the problem recursively until reaching a desired threshold. The method used for coarsening a SAT instance is simple. The algorithm merges pairs of variables chosen randomly to form clusters. In a cluster, the enclosed variables represent the same variable and this way the problem will be easier and hence solved in a coarsening way. The clusters are then used to define a coarser and smaller problem in the next iteration. The remaining variables, which have not been merged, are simply copied to the next level. With this method the problem size is reduced by half each iteration, which means it takes O(log2n) time. This coarsening technique is exact: indeed, at any stage after initialization, the current solution could simply be extended (see extension algorithm bellow) through all the problem levels to achieve a solution of the original problem.

5.2 Initialization Algorithm

For the initialization, we use BeeInit (described in Section 3) to compute the initial solution. BeeInit has to assign values (true or false) to the variables of clusters formed

at the coarsest level. All the variables of the same cluster will be assigned the same value because, as said previously, a cluster represents one variable. In the example from Figure 1, the coarsest level is level 3. At level 3 the algorithm will assign values to the clusters C1 and C2.

C1 = { { {x1, x3}, {x7, x10} },{x2, x4} } value = true, that is value true to x1, x3, x7, x10, x2 and x4
C2= { {x6, x9}, {x5, x8} } value = false, , that is value false to x6, x9, x5 and x8

Thus, all the clauses containing the variables in C1, or the negated form of the variables in C2 will be satisfied. We can see that the problem at level 3 will consider only two variables, this is why it is a smaller and easier version of the initial problem.

5.3 Extension Algorithm

The extension is the reverse procedure of the coarsening. It just unfolds each cluster by placing the clusters that compose it and assigns them the same value assigned to the original cluster. This way, the number of clusters and thus the number of variables will increase. Using the same example as before, to pass from level 3 to level 2 the extension algorithm will return the following situation with three clusters:

C_3 = { $\{x_1, x_3\}$, $\{x_7, x_{10}\}$ } value = true
C_4 = $\{x_2, x_4\}$ value = true
C_2= { $\{x_6, x_9\}$, $\{x_5, x_8\}$ } value = false

This strategy will guarantee that the number of satisfied clauses by the solution will remain the same before and after the extension.

5.4 Refinement Algorithm

The refinement algorithm searches for the best solution of current problems starting with an initial solution extended from the previous level. In this case, the bee swarm optimization as described in Section 3 is used. In the coarse levels, BSO will deal with fewer variables while searching because clusters include variables having the same value. The bees will work on smaller and easier versions of the initial problems until reaching level 0. At this level, BSO will treat the real instance with the whole set of variables but with an interesting initial solution. The gain that is brought from the coarsening and extension processes is to yield a good initial solution that is closer to the optimal one for the problems of the different levels and especially for the original one that corresponds to level 0. In other terms, all the coarsened problems, that are much simpler than the original one, will contribute for searching a solution with good quality to better start BSO when solving the whole instance.

6 Experimental Results

6.1 Benchmark Instances Description

To illustrate the potential gains offered by the multilevel bee swarm optimization, we chose the largest instances of the Sat-encoded Bounded Model Checking (BMC)

problems from the SATLIB (http://www.cs.ubc.ca/~hoos/SATLIB /benchm.html) benchmarks. The following table lists the instances used in our experiments. The second and third columns indicate the number of variables and the number of clauses, respectively.

Table 1. Benchmark instances

Instance	#Vars	#Clauses	Instance	#Vars	#Clauses
bmc-ibm-1	9658	55870	bmc-galileo-9	63624	326999
bmc-ibm-2	3628	11683	bmc-ibm-10	61088	323700
bmc-ibm-3	14930	65728	bmc-ibm-11	32109	150027
bmc-ibm-5	9396	41207	bmc-ibm-12	39598	194778
bmc-ibm-7	8710	39774	bmc-ibm-13	13215	65728
bmc-galileo-8	58074	294821			

6.2 Numerical Results

The tests were carried out on a TOSHIBA machine with 2 GHz CPU and 2 GB of memory and the code was written in Java 1.6. As described in Section 3, BSO has four parameters. Preliminary tests have been carried out in order to fix the values used in the actual experiments:

NBees (Number of bees): 35.
MaxIter (Maximum number of iterations of BSO): 40.
MaxChances (Number of chances): 10.
Flip (Number of flips): 25 flips for each 100 clusters.
Coarsening threshold Experiments showed that we get better results if we stop coarsening when the number of clusters reaches 100, that is solving instances with 100 variables.

Since the shapes of the cuves of all instances are similar, we present in the Figure 1, the progress of the best solution selected from the table dance for both BSO and MLBSO in the first 300 seconds for bmc-galileo-9.cnf. Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

We noted that MLBSO goes through two phases. In the first one, the best solution improves rapidly at first, and then flattens off. It is in this phase that most of the clauses became satisfied. In the second phase the multilevel version behaves more and more like the simple BSO, especially when the finer levels are reached and the number of clusters nears the number of variables of the original problem.

The multilevel approach successively *approximates* the problem with smaller, and hence easier to solve, versions. And since the coarsening in this case is exact, it actually *filters* the solution space by placing restrictions on which solutions the refinement algorithm can visit. Indeed, by changing the value of one cluster in a coarsened space is equivalent to changing the values of several variables in the original solution space thus dealing with several clauses at the same time. In other words, the coarsening strategy allows the possibility of exploring different regions in the search space (diversification) while intensifying the search by exploiting the solutions from previous levels in order to reach better solutions.

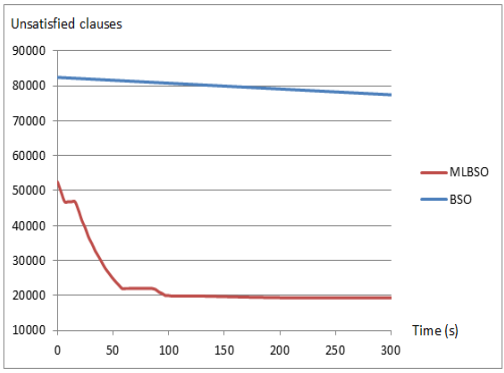
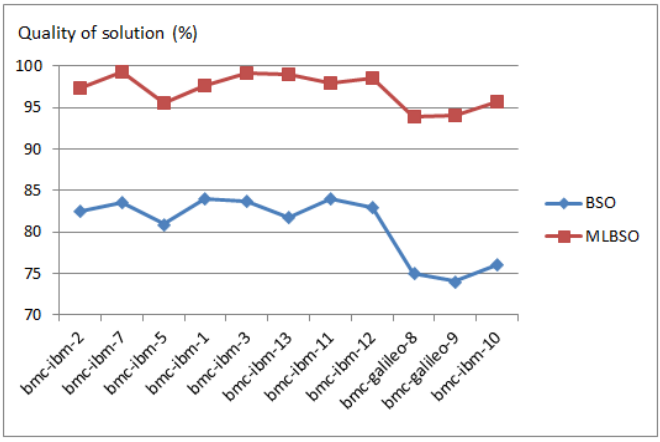
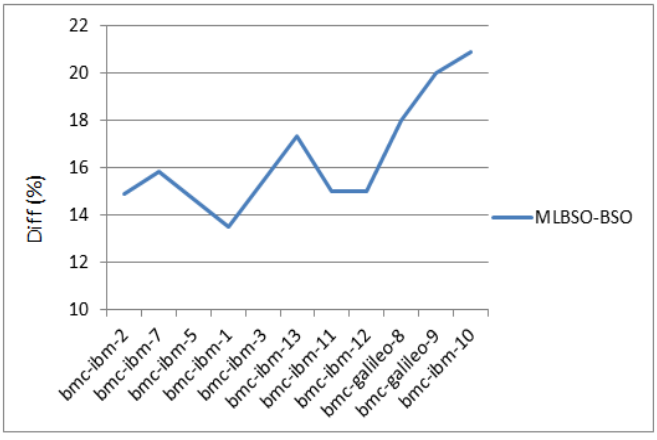


Fig. 1. Results of bmc-galileo-9.cnf



(a)



(b)

Fig. 2. (a) Quality of solution obtained by both methods (b) Difference between the performances of MLBSO and BSO depending on the size of the benchmark

Figure 2 (a), shows the quality of solution reached by both methods after 300 second run time for all benchmark instances which are ordered by increasing size. We can see that the multilevel method managed to keep the quality over 90% for all instances. In Figure 2 (b), we can see that the gap between the performances of MLBSO and BSO becomes wider as the size of the instance increases.

6.3 Performances Comparison

Table 2 represents a comparison between the results (number of the remaining unsatisfied clauses) achieved after 300 sec by our MLBSO and the MLV-MEMETIC (multilevel memetic algorithm) presented in [6]. It shows that for the instances highlighted in grey, the MLBSO outperform the MLV-MEMETIC in terms of quality of solution and run time. As for bmc-ibm-2.cnf, bmc-galileo-8.cnf and bmc-galileo-9.cnf MLV-MEMETIC has a slightly better performance than MLBSO.

Table 2. Comparison between MLV-MEMETIC [6] and MLBSO

Instance	MLV-MEMETIC	MLBSO	Instance	MLV-MEMETIC	MLBSO
bmc-ibm-1	5400	1300	bmc-galileo-9	14500	19300
bmc-ibm-2	200	300	bmc-ibm-10	21500	13700
bmc-ibm-3	4000	550	bmc-ibm-11	14500	3000
bmc-ibm-5	1900	1800	bmc-ibm-12	18000	2800
bmc-ibm-7	2500	250	bmc-ibm-13	4900	650
bmc-galileo-8	12100	17900			

7 Conclusion

In this work, we introduced a new problem solving approach called MLBSO by hybridizing the multilevel paradigm with the Bee Swarm Optimization metaheuristic. The purpose was to enhance BSO and make it capable of handling large SAT problem instances in an efficient manner. The experiments clearly illustrate the important gap that may exist between the performances of both algorithms on very larger instances, which may indicates that the multilevel paradigm is more adapted to handle large scale instances. For future work we plan to investigate further the gain that may be obtained when using the multilevel paradigm for solving harder and larger instances of variant problems.

References

1. Drias, H., Sadeg, S., Yahi, S.: Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 318–325. Springer, Heidelberg (2005)
2. Walshaw, C.: Multilevel Refinement for Combinatorial Optimisation: Boosting Metaheuristic Performance. In: Blum, C., Aguilera, M.J.B., Roli, A., Sampels, M. (eds.) Hybrid Metaheuristics. SCI, vol. 114, pp. 261–289. Springer, Heidelberg (2008)
3. Cook, S.: The Complexity of Theorem-Proving Procedures. In: Proceeding 3rd Annual ACM Symposium on the Theory of Computing, New York, pp. 151–198 (1971)

4. Davis, M., Logemann, G., Loveland, D.: A Machine Program for Theorem Proving. Communications of the ACM 5(7), 394–397 (1962), doi:10.1145/368273.368557
5. Biere, A., Heule, M., Maaren, H.V., Walsh, T.: A history of Satisfiability. In: Handbook of Satisfiability, pp. 3–55. IOS Press, Amsterdan (2009)
6. Bouhmala, N.: A Multilevel Memetic Algorithm for Large SAT-Encoded Problems. Evolutionary Computation 20(4), 641–664 (2012)
7. Karaboga, D., Akay, B.: A survey: algorithms simulating bee swarm intelligence. Artificial Intelligence Review 31(1-4), 61–85 (2009)
8. Blum, C., Puchinger, J., Raidl, G.R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing 11(6), 4135–4151 (2011)

Voting-XCSc: A Consensus Clustering Method via Learning Classifier System

Liqiang Qian[1], Yinghuan Shi[2], Yang Gao[2], and Hujun Yin[3]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
[2] State Key Laboratory for Novel Software and Technology, Nanjing University, China
[3] School of Electrical and Electronic Engineering, The University of Manchester, UK

Abstract. In this article, a novel consensus clustering method (voting-XCSc) via learning classifier system is proposed, which aims (1) to automatically determine the clustering number and (2) to achieve consensus results by reducing the influence coming from the randomness. When conducting the clustering for the data points, the proposed voting-XCSc will first employ the XCSc to generate a set of clustering results with different clustering numbers, and then it will adopt the dissociation-based strategy to experimentally determine the clustering number among all the candidates. Finally, a majority voting-based consensus method is applied to obtain the final clustering results. The proposed voting-XCSc has been evaluated on both the toy examples as well as two real clustering-related applications. i.e, lung cancer image identification, image segmentation. The results demonstrate the voting-XCSc can obtain the superior performance compared with XCSc, K-means, and other state-of-the-arts.

Keywords: Learning Classifier System, Consensus Clustering, Reinforcement Learning.

1 Introduction

Learning Classifier System (LCS), which integrates the reinforcement learning and evolutionary computing, is regarded as one of the most frequently employed rule-based learning tools. Based on the learning framework of LCS, Wilson et al. further developed Extended Classifier System (XCS) [1] which is an accuracy extension of LCS, and XCSR [2] which aims to extend the LCS for real value data processing. The family of LCS, also with XCS and XCSR, has been demonstrated its scalability and efficiency for lots of learning-based problem, e.g., reinforcement learning applications, supervised learning, function approximation as well as general prediction [11][6].

Recently, it is known that a few works have been contributed to investigate performance of LCS (including XCS, XCSR) in unsupervised learning task, i.e., clustering. Tamee *et al.* [4] first introduced the XCS for data clustering tasks, which was demonstrated its superior performance compared with K-means clustering algorithm. The major drawback is that its scalability is limited due to its single-rule representation manner. Instead of the single-rule representation, Shi *et al.* [6] proposed a novel multi-rule representation based XCS clustering method. Each data point is represented by couples of rules, and a newly developed rule merging algorithm is employed by iteratively merging

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 603–610, 2013.

the rules that contain a common data point. Also, another method for XCS clustering named XCSc (Clustering with XCS) [7] is developed. The XCSc also adopts the multi-rule based representation, which initializes each data points by randomly generating several rules. Meanwhile, inspired by the hierarchical strategy used in CHAMELEON algorithm, XCSc adopts a graph-based rule merging approach which is able to deal with complex structure data sets. e.g., spirals and circles.

The previous methods [4][6][7] have demonstrated the promising performance in several XCS clustering tasks. However, there are still several major limitations for the previous works. Firstly, the number of clusters in the previous works are manually predefined. Secondly, the clustering performance is randomly influenced by several factors, e.g. genetic algorithm. To develop a robust and efficient clustering method based on learning classifier system, we propose the voting-XCSc in this article. Voting-XCSc will first employ the XCSc to generate a set of clustering results with different clustering numbers, and then it will adopt the dissociation-based strategy to experimentally determine the clustering number among all the candidates. Finally, a majority voting-based consensus method is applied to obtain the final clustering results. The contribution of this paper can be concluded into the following three folds:

- The voting-XCSc can automatically determine the clustering number.
- The voting-XCSc can reduce the influence coming from the randomness.
- Previous XCS clustering methods are evaluated on the toy examples. We first employ the XCS clustering method into real applications.

2 XCSc

In XCSc, the system will automatically check the whole population ([P]) when the new data comes from the environment. For the new data, XCSc will first create the match set ([M]) via adopting the rule-based representation for each data points. The XCSc calculates the corresponding reward for each rule in the match set, then XCSc will also apply the reinforcement learning strategy to update the parameters within the rules. After several round learning procedure, the XCSc compacts and merges the obtained rules to form the rule clusters. Finally, XCSc will determine the corresponding clusters for each new data. Roughly, the process of XCSc for clustering can be divided into two phases: (1) Learning Mechanisms and Rule Compact, (2) Agglomerative Rule Merging.

3 Voting-XCSc

In this section, we will discuss the technical details of voting-XCSc. The difference between the XCSc and our proposed voting-XCSc can be referred to Fig.1.

Assuming a data set includes N data points which will be clustered into k clusters, for the i^{th} running, we can denote the corresponding clustering results as $\lambda_k^{(i)} \in \mathbb{R}^N$. Also, $\lambda_k^{(i)}(j)$ means the cluster index of the j^{th} data point, which should satisfy $1 \leq \lambda_k^{(i)}(j) \leq k$.

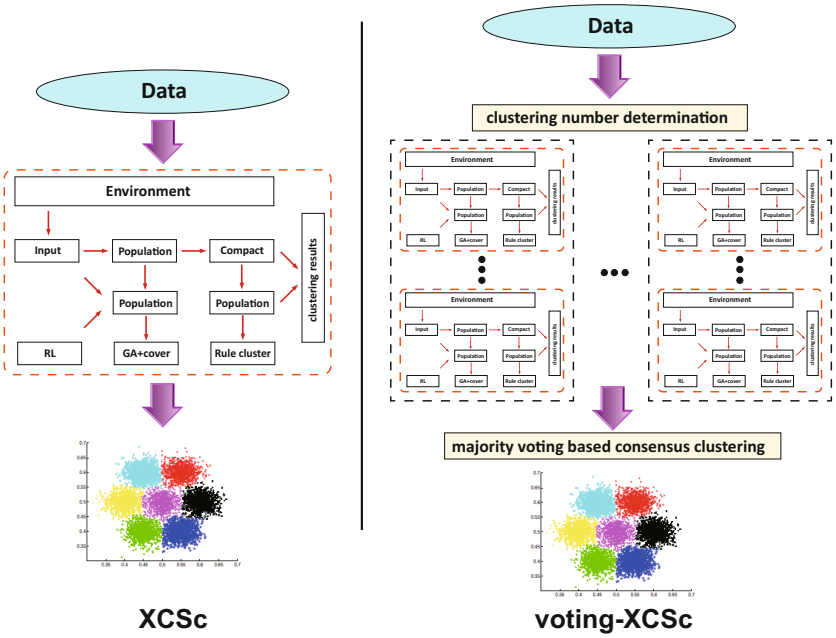


Fig. 1. The difference of XCSc and voting-XCSc

3.1 Consistency Validation

Different clustering results might be different even they are actually equivalent. To avoid this situation, we introduce the consistency validation strategy proposed in ref, which can be mathematically defined as

$$\begin{cases} \lambda_k^{(i)}(1) = 1 \\ \lambda_k^{(i)}(j+1) \leq \arg\max_{m=1,\ldots,j} \lambda_k^{(i)}(m) + 1 \end{cases} \tag{1}$$

Thus, for each clustering result obtained by employing XCSc, we will first conduct the consistency validation.

3.2 Determining the Clusters Number

We employ the distortion theory based strategy [3] to automatically determine the clusters number. Since for a certain cluster number, we have several clustering results via running XCSc repeatedly, the corresponding minimum average distortion can be calculated as follows:

$$MAD = \frac{1}{d * Q} \min_{c_1,\ldots,c_k} E[(X - c_x)^\top \Sigma^{-1}(X - c_x)], \tag{2}$$

where $c_1, ..., c_k$ indicates the k clustering centers, and c_x means the clustering center which is closed to X. d is the dimension of the data point, Q is the number of clustering results under clustering number k. Once we calculated the minimum average distortion for each different clustering number k, we can adopt the Jump algorithm [3] to get the optimal clustering number k_{opt}.

3.3 Majority-Voting Based Consensus Clustering

When obtaining the k_{opt}, the corresponding clustering results will be selected. For the purpose of consensus clustering, we employ the majority voting as follows:

$$\lambda^* \leftarrow \texttt{majority voting}\{\lambda_{k_{opt}}^{(1)}, \lambda_{k_{opt}}^{(2)}, ..., \lambda_{k_{opt}}^{(Q)}\} \tag{3}$$

Finally, we give the algorithm details of the proposed voting-XCSc as in Algorithm.1.

Algorithm 1. Voting-XCSc

Input: K_{max}, Q (K_{max} is the predefined maximum clustering number)
Output: λ^*
1: **for** $k \leftarrow 1, ..., K_{max}$ **do**
2: **for** $i \leftarrow 1, ..., Q$ **do**
3: $\lambda_k^{(i)} \leftarrow$ run XCSc with k clustering number
4: $\lambda_k^{(i)} \leftarrow$ consistency check by Eq.1.
5: **end for**
6: **end for**
7: $k_{opt} \leftarrow$ returned by Jump algorithm [3].
8: $\lambda^* \leftarrow$ returned by Eq.3.

4 Experimental Results

We evaluated the proposed voting-XCSc on both the toy examples and two real applications (lung cancer image identification, image segmentation), which aims to use the clustering algorithm for different tasks.

4.1 Settings

For the parameters setting in the experiment, we follow the same values of the corresponding parameters in XCSc [7] for fair comparison. The maximum rule number of the population is 1000, the round number of training stage is 2000-20000, the learning rate is set to 0.2, the probability of mutation and crossover in GA are set to 0.001 and 0.8, respectively. The parameters α and v used to calculate the accuracy are set to 0.1 and 5, respectively.

4.2 Toy Example

We use the two-Gaussian and two-Moon data sets: two-Gaussian data set includes 2000 data points which are randomly generated from the two predefined Gaussian distribution; two-Moon data set also includes 2000 data points which are randomly generated from two semi circles. The two data sets can be refereed to Fig.2.

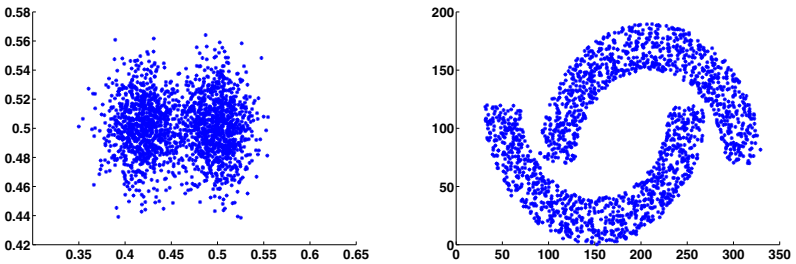


Fig. 2. Examples of two toy data sets

Since the labels of the two data sets are known, we can evaluate the proposed voting-XCSc by comparing with related methods: K-means and XCSc [7]. Also, to eliminate the influence of randomness during data set generation, we repeat the experiment for 100 times. The results of classification accuracy can be referred to Fig.3. It is obvious that voting XCSc can outperform K-means and XCSc in terms of higher classification accuracy.

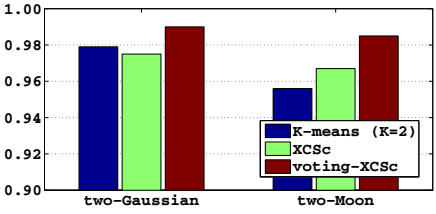


Fig. 3. Classification accuracy compared with related methods

4.3 Lung Cancer Image Identification

Lung Cancer Image Identification aims to automatically classify the images into one of the following five categories: normal (NC), squamous carcinoma (SC), adenocarcinoma (AC), small cell cancer (SCC) and nuclear atypia (NA). In [5], Shi *et al.* proposed a novel transductive cost-sensitive lung caner image identification method, which applies the K-means algorithm to learn the codebook. Since voting-XCSc can also be employed for codebook learning (clustering), we test the classification ability of the codebook generated by using voting-XCSc. The lung cancer image set includes the 271 images, whose typical samples can be referred to Fig.4.

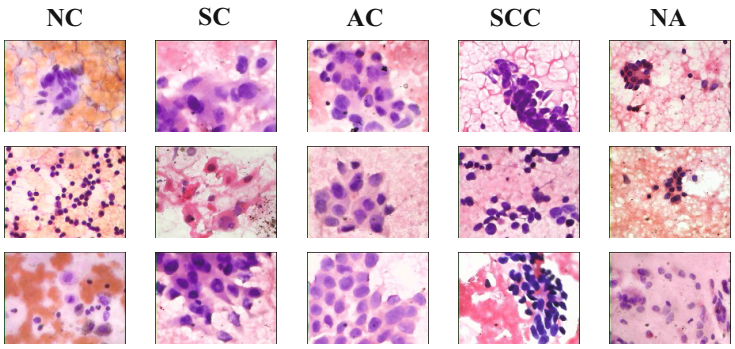


Fig. 4. Typical examples of lung cancer images

For the parameters setting, we uniformly follow all the settings in [5]. Also, we introduced the supervised learning algorithms (kNN, MCMI-AB [8], mcSVM [10]), semi-supervised learning algorithm (LapRLS [9]), transductive cost-sensitive learning algorithms (mCLRLS$_{\mathbf{H}}$ and mCLRLS$_{\mathbf{E}}$) for evaluation. All the methods are evaluated by using 10-fold cross validation (employing 10% and 30% images as training samples. respectively). From the Table.1, we can found that voting-XCSc can achieve the superior or comparable performance compared with the state-of-the-art methods. Err$_m$ means the percentage of misclassifying cancerous images into normal ones, while the costs$_m$ measures the total misclassification costs. Both Err$_m$ and costs$_m$ are the lower the better.

Table 1. Cancerous types classification

Methods(10%)	Err$_m$ %	costs$_m$	Methods(30%)	Err$_m$ %	costs$_m$
kNN	12.89 ± 1.86	510.20 ± 66.52	kNN	10.00 ± 3.56	336.00 ± 78.42
MCMI-AB	4.92	309.00	MCMI-AB	4.61	204.00
LapRLS	13.81 ± 2.23	506.80 ± 64.37	LapRLS	5.66 ± 1.36	232.00 ± 31.62
mcSVM	8.35 ± 4.62	389.20 ± 120.42	mcSVM	7.11 ± 2.52	277.00 ± 42.91
mCLRLS$_{\mathbf{E}}$	1.96 ± 1.52	227.00 ± 42.20	mCLRLS$_{\mathbf{E}}$	1.18 ± 0.86	160.40 ± 31.55
mCLRLS$_{\mathbf{H}}$	2.89 ± 1.61	276.00 ± 83.41	mCLRLS$_{\mathbf{H}}$	1.05 ± 1.44	162.00 ± 11.11
voting-XCSc	2.13 ± 1.41	232.00 ± 62.50	voting-XCSc	0.93 ± 1.21	154.00 ± 14.33

4.4 Image Segmentation

Image segmentation is one of the most important research issues in computer vision. The most commonly-used strategy for image segmentation is clustering. Therefore, we wish to evaluate the performance of the proposed voting-XCSc on segmenting the real images. The images used for segmentation are publicly available from the 'The Berkeley Segmentation Dataset' [12], which includes 300 color images. Compared with K-means clustering algorithm, and XCSc on image segmentation, the advantage is of out voting-XCSc is that the number of clusters is experimentally determined, which does not need the predefinition. We illustrated several typical examples in Fig.5.

Fig. 5. Typical image segmentation results

5 Conclusion

We present a novel consensus clustering method (voting-XCSc) via learning classifier system in this paper. The goal of voting-XCSc can be concluded as (1) to automatically determine the clustering number and (2) to achieve consensus results by reducing the influence coming from the randomness. When conducting the clustering for the data points, the proposed voting-XCSc will first employ the XCSc to generate a set of clustering results with different clustering numbers, and then it will adopt the dissociation-based strategy to experimentally determine the clustering number among all the candidates. Finally, a majority voting-based consensus method is applied to obtain the final clustering results. We successfully apply the proposed voting-XCSc to the toy examples as well as two real clustering-related applications, which validates the effectiveness and efficiency of our method.

Acknowledgement. We would like to acknowledge the support for this work from the National Science Foundation of China (Grant Nos. 61035003, 61175042, 61021062), the National 973 Program of China (Grant No. 2009CB320702), the 973 Program of Jiangsu, China (Grant No. BK2011005) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

References

1. Wilson, S.W.: Classifier fitness based on accuracy. Evolutionary Computation 3(2), 149–175 (1995)
2. Wilson, S.W.: Get real! XCS with continuous-valued inputs. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) IWLCS 1999. LNCS (LNAI), vol. 1813, pp. 209–219. Springer, Heidelberg (2000)
3. Sugar, C., James, G.: Finding the number of clusters in a data set: An information theoretic approach. Journal of the American Statistical Association 98, 750–763 (2003)
4. Tamee, K., Bull, L., Pinngern, O.: Towards clustering with xcs. In: Proceedings of the 9th Genetic and Evolutionary Computation Conference, pp. 1854–1860 (2007)
5. Shi, Y., Gao, Y., Wang, R., Zhang, Y., Wang, D.: Transductive Cost-Sensitive Lung Cancer Image Classification. Applied Intelligence 38, 16–28 (2013)
6. Shi, L., Gao, Y., Wu, L., Shang, L.: Clustering with XCS on complex structure dataset. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 489–499. Springer, Heidelberg (2008)
7. Shi, L., Shi, Y., Gao, Y.: Clustering with XCS and Agglomerative Rule Merging. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 242–250. Springer, Heidelberg (2009)
8. Zhu, L., Zhao, B., Gao, Y.: Multi-class multi-instance learning approach for lung cancer cell classi? cation based on bag feature selection. In: Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 487–492 (2008)
9. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res., 7:2399–7:2434 (2006)
10. Morik, K., Brochhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring. In: Proceedings of the 16th International Conference on Machine Learning (ICML), pp. 268–277 (2009)
11. Butz, M.V.: Learning classifier systems. In: Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation (2007)
12. The Berkeley Segmentation Dataset and Benchmark, `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`

Distance Weighted Cosine Similarity Measure for Text Classification

Baoli Li and Liping Han

Department of Computer Science
Henan University of Technology
1 Lotus Street, High & New Industrial Development Zone
Zhengzhou, Henan 450001, China
csblli@gmail.com

Abstract. In Vector Space Model, Cosine is widely used to measure the similarity between two vectors. Its calculation is very efficient, especially for sparse vectors, as only the non-zero dimensions need to be considered. As a fundamental component, cosine similarity has been applied in solving different text mining problems, such as text classification, text summarization, information retrieval, question answering, and so on. Although it is popular, the cosine similarity does have some problems. Starting with a few synthetic samples, we demonstrate some problems of cosine similarity: it is overly biased by features of higher values and does not care much about how many features two vectors share. A distance weighted cosine similarity metric is thus proposed. Extensive experiments on text classification exhibit the effectiveness of the proposed metric.

1 Introduction

Similarity calculation is a basic component for many text mining applications. For example, if we have a perfect method to assess how two text segments are similar, we could build an ideal information retrieval system. In the past years, a lot of metrics [1,2], such as Euclidean distance based metric, Cosine, Jaccard, Dice, Jensen-Shannon Divergence based metric, have been proposed to deal with different kinds of information retrieval and natural language processing problems. Among the existing metrics, Cosine, which measures the angle between two vectors, is the most popular one. It is effectively calculated as dot-product of two normalized vectors.

Given two *N* dimension vectors $\vec{v}$ and $\vec{w}$, the cosine similarity between them is calculated as follows:

$$\text{Cosine}(\vec{v},\vec{w}) = \frac{\vec{v} \bullet \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

In mathematics perspective, Cosine similarity is perfect. However, if we check it in text mining perspective, it may not always be reasonable. Let's consider a few example vectors shown in figure 1. Suppose these 3-D vectors are derived from five text segments

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 611–618, 2013.

A, B, C, D, and E. The cosine similarities between segment A and the rest are given in the figure. From the values, we can conclude that segment B is the most similar one of A, as it has the highest cosine. However, is it reasonable? Intuitively, text segments C and E, which both have two common terms with segment A, are more relevant to A than B, which contains only one term. Moreover, the segment E has one more term than segment A, but Cosine(A,E) is much lower than Cosine(A,B). If we regard the additional term as a noise and neglect it, E will have the same vector as A.

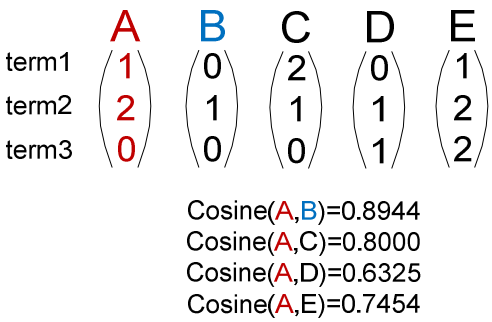


Fig. 1. Cosine similarities between five synthetic vectors

It can thus be derived from the above figure that cosine similarity tends to be overly biased by the features of higher values, but it doesn't care much about how many features two vectors share. In text mining perspective, more features two text segments share, more similar they are. If a part of a text segment is much similar to another segment as whole, the former one is usually thought to be relevant to the latter in Information Retrieval. It is this observation that motivates us to explore more effective similarity metrics than Cosine for text mining.

Because of the proven effectiveness of cosine similarity, we decide to derive new metrics by slightly modifying it. Several distance weighted versions are explored, where distance tends to capture how many features two text segments share. With extensive experiments on a classical text mining problem, i.e. text classification, we obtain a distance weighted cosine metric that performs better than the original cosine metric in most cases. It is also demonstrated with experiments that the similarity metric does have important effects in text mining applications.

The rest of this paper is organized as follows: section 2 introduces the explored distance weighed cosine metrics; section 3 presents extensive experiments on three text classification problems and discussion on the results; Section 4 concludes the paper.

2 Distance Weighted Cosine Similarity Measures

We explore different new similarity metrics with the following evidences or assumptions: 1. cosine similarity is good enough for most text mining applications; 2. more features two text segments share, more similar they are. Therefore, all of the designed metrics have two key components: cosine similarity and distance measure, but they are different in applying different distance measures and assembling strategies.

The explored distance measures, which are expected to capture how many features two vectors share or not, include:

***Hamming Distance*:** it counts how many features two vectors do not share. As vectors may have quite different numbers of valid features, a normalized version, which stands for the percentage of distinct features, is used. Given two N-dimension vectors $\vec{v}$ and $\vec{w}$, a possible formula to calculate this measure is as follows:

$$HD(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N}(\mathrm{sgn}(v_i) - \mathrm{sgn}(v_i * w_i))}{\sum_{i=1}^{N}\mathrm{sgn}(v_i)} + \frac{\sum_{i=1}^{N}(\mathrm{sgn}(w_i) - \mathrm{sgn}(w_i * v_i))}{\sum_{i=1}^{N}\mathrm{sgn}(w_i)},$$

while sgn(x) is the sign function.

***Weighted Hamming Distance*:** the former measure takes each feature equally important, which is not ideal. A simple improvement is to weight counts with features' values as follows:

$$WHD(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} v_i * (1 - \mathrm{sgn}(v_i * w_i))}{\sum_{i=1}^{N} v_i} + \frac{\sum_{i=1}^{N} w_i * (1 - \mathrm{sgn}(w_i * v_i))}{\sum_{i=1}^{N} w_i}.$$

To assemble ***Cosine*** similarity (dubbed ***Cosine***) and ***Dist***ance measure (dubbed ***Dist***), several strategies can be applied. As ***Cosine*** and ***Dist*** are inverse, we firstly turn the ***Dist*** into some similarity format, e.g. $\frac{1}{Dist+1}$, and then assemble them together with multiplying, averaging, or other strategies. Some possible strategies include:

Multiplying: $\frac{Cosine}{Dist+1}$.

Averaging: $\frac{1}{2} * (Cosine + \frac{1}{Dist+1})$ or $\frac{2 * \frac{Cosine}{Dist+1}}{(Cosine + \frac{1}{Dist+1})}$.

All the above formulas can have different variants. For example, we can add some constants like 1/2, give different weights for the two similarities, and use power or log function to reduce the influence of one component. With extensive experiments on text classification problems, we find the following distance weighted cosine metric could achieve better performance than the traditional cosine similarity in almost all the cases.

$$Dw - Cosine = \frac{1}{Dist^2 / Cosine + 1} = \frac{Cosine}{Dist^2 + Cosine}$$

The weighted hamming distance measure is used in the above formula.

3 Experiments and Discussion

In order to evaluate the performance of different distance weighted cosine metrics, we conduct extensive experiments on three single-label text classification problems[3].

3.1 Datasets

We experiment with the following three datasets:

***20 Newsgroups*:** this dataset is evenly partitioned into 20 different newsgroups, each corresponding to a specific topic. Its "bydate" version is widely used in literature, as it has a standard training and test split. The training set has 11,293 samples and the test set 7,528 samples.

***Reuters52c*:** it is a single-label dataset derived from Reuters-21578 with 90 classes by Ana Cardoso-Cachopo during her Ph.D. study [4]. Documents with multiple labels in the original Reuters-21578 (90 classes) dataset are discarded and finally the Reuters52c dataset contains 52 categories, 6,532 documents for training, and 2,568 documents for test. The dataset is imbalanced and some categories only have a few documents, e.g. classes *cpu* and *potato*. We use the all-terms version without stemming.

***Sector*:** this dataset is a collection of web pages belonging to companies from various economic sectors. It has 104 categories, 6,412 training samples, and 3,207 test samples.

3.2 Experimental Settings

We use the vector space model (VSM) for data representation, in which the dimension is determined by the size of the dataset's vocabulary. Each document is then represented as a space vector where the words in the document are mapped onto the corresponding coordinates. In the feature-selection phase of the experiments, we removed words that occur only once [5]. The weight of a feature is given as follows:

$$x_i = \frac{(1+\log(TF(w_i,d)))\cdot\log(\frac{|D|}{DF(w_i)})}{\sqrt{\sum_j((1+\log(TF(w_j,d)))\cdot\log(\frac{|D|}{DF(w_j)}))^2}},$$

which is the same as the standard representation "ltc" in Manning and Schutze [6]. Here, D is the document collection, and TF and DF are a term's frequency in a document d and its document frequency in the collection D respectively.

In classification, we use two widely used algorithms: Centroid and k-Nearest Neighbor [7, 8]. They are all heavily dependent on similarity metrics. With Centroid algorithm, each category is represented by a centroid vector, and a test sample is then classified to the category that has the highest similarity value between its centroid vector and the test sample's vector. With k-Nearest Neighbor algorithm, the category

prediction of a test sample is made according to the category distribution among the top *k* most similar samples in the training set, where a similarity metric is used to find these *k* Nearest Neighbors.

We evaluate different similarity metrics, including our proposed distance weighted cosine similarity, the original cosine similarity, Jaccard, and others.

3.3 Evaluation Metric

To evaluate the effectiveness of category assignments to documents by classifiers, the harmonic average of the standard precision and recall, F1 measure, is used as follows:

$$\mathrm{F1} = \frac{2recall * precision}{recall + precision}$$

The overall performance on all categories can be computed either by the micro-averaging method or by the macro-averaging method. In micro-averaging, the MicF1 score is computed globally over all the binary decisions. In macro-averaging, the MacF1 score is computed for the binary decisions on each individual category first and then averaged over the categories. The micro-averaged score tends to be dominated by the classifier's performance on common categories, while the macro-averaged score is more influenced by the performance on rare categories.

Table 1. With Centroid classification algorithm, system performance on three datasets with different similarity metrics

Similarity Metric	20 Newsgroups		Reuters52c		Sector	
	MicF1	MacF1	MicF1	MacF1	MicF1	MacF1
Dw-cosine	**84.3916**	**83.6943**	**90.3816**	71.4916	**89.0864**	**89.1664**
Cosine	81.6286	80.8969	89.0187	**71.7848**	87.3402	87.6430
Jaccard	74.5218	73.7537	69.1978	46.2682	76.3018	77.5525

3.4 Results and Discussion

As mentioned in section 2, we conduct extensive experiments with different variants of distance weighted cosine metrics, and find the Dw-Cosine metric performs best. Here we present only the results of this proposed metric from this family. We also experiment with other metrics, e.g Dice and Jensen-Shannon Divergence based metric, but Dice performs similar to Jaccard, and Jensen-Shannon Divergence poorer than Jaccard. So we do not report these metrics' performance here.

Table 1 shows the system performance on three datasets with Centroid classification algorithm. Cosine performs much better than Jaccard on three datasets, and Dw-cosine can increase Cosine's MicF1 with 1.3% to 2.7%. As to the MacF1, Dw-cosine achieves equally good result as Cosine on the Reuters52c dataset, but beats Cosine on the other two datasets with around 2.8% (20 Newsgroups) and 1.5% (Sector), respectively. The improvement is impressive, and it can be seen that choosing a suitable similarity metric is quite important.

To better understand the performance of the three metrics, we depict a column graph in figure 2. Dw-cosine wins the competitions on 5 of 6 tracks, and gains the most on the 20 newsgroups dataset.

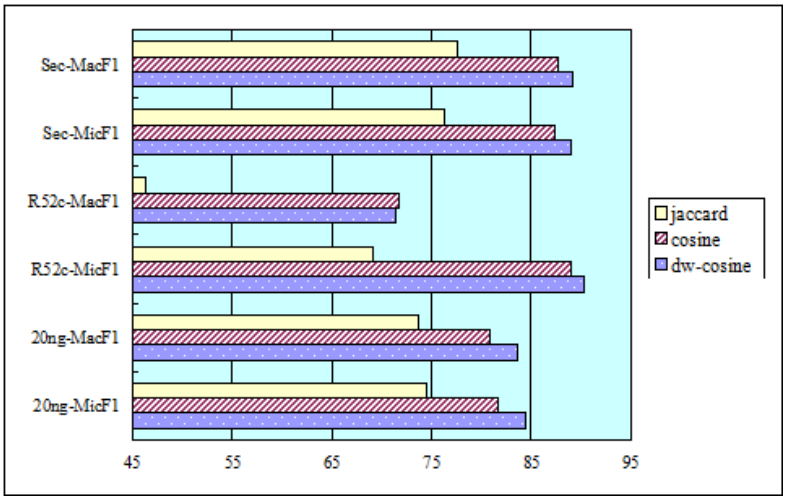


Fig. 2. Visualized system performance on three datasets with Centroid algorithm

Table 2. With k Nearest Neighbour classification algorithm, system performance on three datasets with different similarity metrics, and the values are average of 40 runs (k from 1 to 40)

Similarity Metric	20 Newsgroups		Reuters52c		Sector	
	MicF1	MacF1	MicF1	MacF1	MicF1	MacF1
Dw-cosine	**77.1018**	**76.5166**	88.4940	**68.2012**	**84.8566**	**84.6625**
Cosine	76.1351	75.5760	86.6715	68.1401	82.6458	82.4599
Jaccard	71.4865	71.0726	**89.5113**	63.3028	75.5862	75.3666

Table 2 gives the three metrics' performance with k-Nearest Neighbor algorithm. We vary k from 1 to 40, and then the values in the table are average of these 40 runs. Similar to table 1, Cosine beats Jaccard on 5 tracks, but obtains much poorer MicF1 value on the Reuters52c dataset than Jaccard, which also indicate how important a suitable similarity metric is for a specific text mining problem. Dw-cosine exhibits consistent advantages over Cosine similarity with k-Nearest Neighbor algorithm. It can boost the MicF1 and MacF1 of Cosine from 1% to 2% except for MacF1 on Reuters52c.

T-tests over the 40 runs show that the performance differences in table 2 are all significant but for MacF1 scores of Dw-cosine and Cosine on the Reuters52c dataset.

Similarly, we present a column graph to visualize how these three metrics perform with k-Nearest Neighbor algorithm in figure 3. Dw-cosine and Cosine achieve almost the same MacF1 scores on the Reuters52c dataset, while Dw-cosine shows its biggest advantages over Cosine on the Sector dataset.

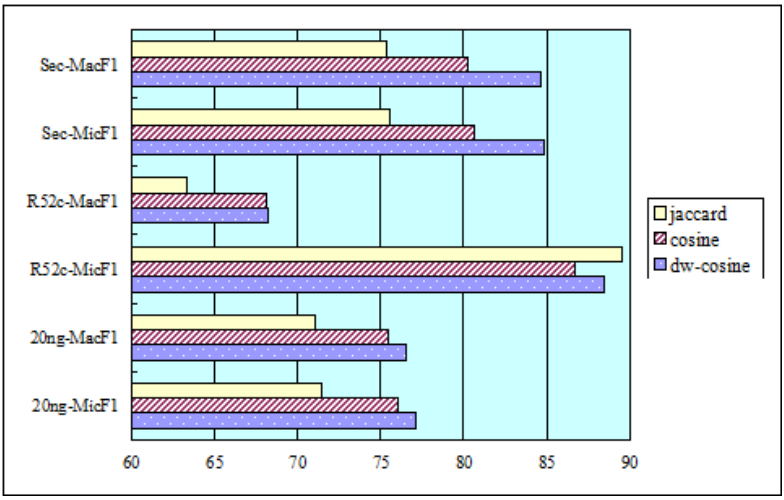


Fig. 3. Visualized average system performance on three datasets with kNN algorithm

4 Conclusions and Future Work

Cosine similarity is widely used in information retrieval, natural language processing, and text mining. It's quite effective, but not perfect. We demonstrate with a few synthetic examples that Cosine similarity tends to be biased by features of higher values and not to pay enough attention to how many features two vectors share. It is this observation that motivates this study. A family of distance weighted cosine metrics is explored and a specific one of this family, i.e. Dw-cosine, achieves consistently better results than the original cosine similarity on three text classification problems. Our experiments also demonstrate that similarity metric may have critical impacts on system performance.

In the future, we would like to experiment with more datasets and apply the proposed distance weighted cosine similarity into other text mining applications, e.g. information retrieval, word sense disambiguation, and so on. We also plan to explore how to detect dataset's properties and suggest suitable similarity metric automatically.

Acknowledgments. This work is supported by the Henan Provincial Research Program on Fundamental and Cutting-Edge Technologies (No. 112300410007), and the High-level Talent Foundation of Henan University of Technology (No. 2012BS027).

References

1. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. (1989) Addison-Wesley Longman Publishing, Boston, MA.

2. Li M., Chen X., Li X., Ma B., and Vitanyi P. M.B. : The Similarity Metric. IEEE Transactions on Information Theory, 50(12): 3250-3264 (2004)
3. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34, 1 (2002), 1-47.
4. Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Técnico, Portugal (2007)
5. Yang Y. and Pedersen J. O.: A comparative study on feature selection in text categorization. In Proceedings of Fourteenth International Conference on Machine Learning. (1997): 412-420.
6. Manning C. D. and Schutze H. 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.
7. Yang Y. and Liu X.: A re-examination of text categorization methods. In Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1999): 42-49.
8. Li B., Lu Q., and Yu S.: An adaptive k-nearest neighbor text categorization strategy. ACM Transactions on Asian Language Information Processing (TALIP) 3.4 (2004): 215-226.

A Survey on Benchmarks for Big Data and Some More Considerations

Xiongpai Qin[1] and Xiaoyun Zhou[2]

[1] School of Information, Renmin University of China, Beijing, 100872, China
[2] Computer Science Department, Jiangsu Normal University, Xuzhou, Jiangsu, 221116, China
qxp1990@sina.com, qxp1990@ruc.edu.cn, wsp0516@163.com

Abstract. A big data benchmark suite is needed eagerly by customers, industry and academia recently. A number of prominent works in last several years are reviewed, their characteristics are introduced and shortcomings are analyzed. The authors also provide some suggestions on building the expected benchmark, including: component based benchmarks as well as end-to-end benchmarks should be used together to test distinct tools and test the system as a whole; workloads should be enriched with complex analytics to encompass different application scenarios; metrics other than performance metrics should also be considered.

Keywords: Big Data, Benchmark, Performance, Scalability, Fault Tolerance, Energy Saving, Security.

1 Introduction

With the going down of the price of storage devices, development of e-commerce and internet applications such as Twitter and Facebook, requirements of science research etc., people have collected some data sets of huge volumes that haven't been seen before. Data comes in different forms (structured and unstructured), with different velocities (some data are generated automatically by sensors which needs to be processed timely). RDBMS (relational database management systems) can handle structured relational data well, although some unstructured data could be stored in RDBMS for later processing, RDBMS could not address all data management and processing challenges due to its scalability and other limitations. Various noSQL technologies are proposed and implemented in recent years for big data management and processing.

A benchmark is important in that it not only helps customers evaluate similar products from different vendors, but also it is needed by vendors and researchers. Vendors could continuously improve their products by benchmarking. Researchers could try new techniques and use the benchmark to evaluate the techniques, which will facilitate the process of innovation. To select among new technologies in conjunction with RDBMS for efficiently processing of big data, customers, industry as well as academia need a new benchmark [1] [2] [3].

For decades, relational database has been the standard choice for data storage and data analytics in OLTP and OLAP applications. Industries, together with academia

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 619–627, 2013.

have developed benchmark suits to evaluate products from vendors on a fair basis. The most prominent ones are the TPC-C benchmark and the TPC-H benchmark, for OLTP and OLAP scenarios respectively. Taking TPC-H as an example, it is a benchmark of TPC (Transaction Processing Performance Council) for decision support systems. It consists of industry-related data with 8 tables and 22 business-oriented queries. The performance of TPC-H is measured by *QphH@Size* (Query-per-hour at different data sizes). Performance of different systems can be evaluated on the same size datasets. Many database systems running on various hardware platforms are compared using the metrics of *QphH@Size* and *Price/QphH*. Besides TPC-C and TPC-H, TPC-W is a benchmark for evaluation of OLTP-style web applications, TPC-DS and SSB (Star Schema Benchmark) are analytic benchmarks for OLAP and decision support systems.

TPC style benchmarks are kept static, which allows benchmarks to remain comparable across several decades. In the mean time, it has also incurred some criticism that the benchmarks are stale and not enough for benchmarking of recent big data applications, since big data is likely to evolve over time.

2 Some Recent Efforts

2.1 Survey of Recent Efforts

MapReduce has risen to become the de-facto tool for big data processing. Kim et. al. designed MRBench [4] for evaluating the MapReduce framework using TPC-H workloads. They implemented the needed relational operations, including selection, projection, cross product, join, sort, grouping, and aggregation. They have also conducted experiments by varying data sizes, number of nodes, and number of map tasks. MRBench could give people some insights into decisions on using MapReduce for data warehouse style queries. However some big data applications have gone beyond simple SQL queries to deeper analytics.

A use case that comprises five representative queries for massive astrophysical simulations is proposed in [5]. The authors have implemented the use case in a distributed DBMS and in a Pig/Hadoop system. The performances of the two platforms are compared against each other. They found that both systems provide competitive performance and improved scalability relative to IDL-based methods. However, the use case could not be a fully functional benchmark in itself, because it only considers astrophysical applications, and the queries of the use case could not represent workloads from other domains.

PUMA [7] is a benchmark suite proposed specifically for MapReduce. The workload of the benchmark includes term-vector, inverted-index, self-join, adjacency-list, k-means, classification, histogram-movies, histogram-ratings, sequence-count, ranked inverted index, Tera-sort, GREP, and word-count. There are some shortcomings for MapReduce specific benchmarks. While the application domains currently dominated by MapReduce should be a part of big data benchmarks, the existing MapReduce benchmarks aren't complete representatives of all big data systems. They are inherently limited to measuring only a stripe of the full range of system behavior, since a real life cluster hardly runs only one job at a time or just a

handful of some specific jobs. Benchmarks such as [6] and [8] are also designed specifically for MapReduce, they hold the shortcomings mentioned above.

In the work of [9], the authors discuss many aspects of building a new benchmark for big data. In their opinion, a unified model should be used, which could describe several components of the targeted workloads, including the functions that the system must hold, the representative data access patterns, the scheduling and load variations over time, and the computation (basic operations, queries, analytic algorithms etc.) required to achieve the functional goal.

They have laid down some standards for the expected benchmark, including: (1) *representative*: the benchmark should measure performance using metrics that are relevant to real life application domains, and under situations that similar to real life computing scenarios. (2) *Portable*: the benchmark could be ported to different kinds of systems that can serve the computing needs of the same application domain. (3) *Scalable*: the benchmark could measure performance for both large and small systems, and for both large and small data sizes. (4) S*imple*: the benchmark could be easily understood. (5) S*ystem diversity*: big data systems store and manipulate many kinds of data. Sometimes diversity translates to mutually exclusive variations in system design. (6) R*apid data evolution*: analytic needs evolve constantly and rapidly. The system should be able to scale at multiple layers, and the data should be able to scale across multiple sources and formats.

The authors believed that identifying the functions of big data application domains is the first step toward building truly representative big data benchmark. Big data encompasses many application domains. OLTP is only one domain, thus application domains should also be included, i.e. (1) *flexible latency analytics*, for which MapReduce was originally designed, (2) *interactive analytics with low latency* like traditional OLAP, (3) *semi-streaming analytics*, which describes continuous computing processes.

Yahoo! Cloud Serving Benchmark (YCSB) [10] is proposed with the goal of facilitating performance comparisons of the new generation of cloud data serving systems. Examples of systems for cloud serving include BigTable, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and many others noSQL databases. Considering the scaling out architecture, the elasticity, and the high availability characteristics of a cloud, the authors have discussed some trade-offs made by cloud data systems, including read performance vs. write performance, latency vs. durability, synchronous vs. asynchronous replication, and data partitioning. The YCSB benchmark evaluates the target system from three tiers, i.e. the performance tier, the scaling tier, and the availability tier. A core set of functions are defined and experiment results of four widely used systems, i.e. Cassandra, HBase, Yahoo's PNUTS, and a simple sharded MySQL, are reported. The benchmark addresses the problem of benchmarking "cloud based OLTP" systems, which typically do not support ACID transactions.

Smullen et. al. [11] believed that a large fraction of the data that will be stored and processed in future systems is expected to be unstructured, in the form of texts, images, audio etc. They presented an unstructured data processing benchmark suite. Detailed descriptions of the workloads are provided, including edge detection, proximity search, data scanning, and data fusion. The workloads could capture larger space of application characteristics. Although the benchmark only aims at

unstructured data processing, it could be a component in a big data benchmark for evaluation of the target system specifically in the aspect.

In the research of Huang and his colleagues [12], HiBench which is a benchmark specifically for MapReduce framework evaluation, is introduced. The authors have analyzed some previous benchmarks. Sorting benchmarks [13] such as TeraSort [14] are used by Google and Yahoo [15] to test MapReduce optimizations. However sorting is only one aspect of data processing although it is basically important. GridMix [16] is a synthetic benchmark in the Hadoop distribution, which models data-access patterns of a Hadoop cluster by running a mix of Hadoop jobs, including 3-stage chained MapReduce job, large data sort, reference select, indirect read, and text sort. GridMix is not a fully-blown benchmark for big data since its workloads are not representative enough to include other important characteristics of real world applications. The Hive performance benchmark [17] is used to compare the performance of Hadoop and parallel analytical database, which consists of five programs with GREP as the first one, the other four analytical queries are designed for traditional structured data analysis [18], including selection, aggregation, join and UDF aggregation. The DFSIO [19] is a file system level benchmark for Hadoop that tests how HDFS handles a large number of tasks performing writes or reads simultaneously. Based on their analysis, finally the authors propose HiBench [12] to accommodate several components, which make it different from other benchmarks, including micro benchmarks - sort, word count and TeraSort; Web search - Nutch indexing and page rank; machine learning - the Bayesian classification and K-means clustering; HDFS benchmark - a file system level benchmark. Extensive experimental results using their benchmark are provided. The most prominent characteristic of the benchmark is that it incorporates machine learning algorithms into the whole framework, which is coincident with the trend of complex analysis over the big data.

Zhan et. al. have a strong background of HPC (high performance computing), they proposed CloudRank-D [20] for benchmarking cloud computing system for data processing. The benchmark suite includes basic operations for data analysis, classification, clustering, recommendation, sequence learning, association rule mining, and data warehouse queries. The benchmark suite also includes a representative application - ProfSearch (http://prof.ict.ac.cn/). Two simple metrics: data processed per second and data processed per Joule are used as two complementary metrics for evaluating cloud computing systems specifically. The metric of data processed per Joule is defined as the total amount of data inputs of all jobs divided by the total energy consumed during the duration from the submission time of the first job to the finished time of the last job. Inclusion of energy consumption metrics makes the benchmark different from others.

Researchers from industry and academia gathered frequently [21] in recent two years, and considered the building of a truly new and comprehensive benchmark for big data. They discussed almost every aspects of big data benchmark, including representative workloads, data sets, new metrics, implementation options, tool set, data generation, benchmark execution rules, and the specification with scaling factor etc. The SIGMOD paper of [22] presented the BigBench, an end-to-end big data benchmark proposal. The proposal covers data models of structured, semi-structured and unstructured data, and addresses variety, velocity and volume aspects of big data systems. The structured part of the BigBench data model is borrowed from the TPC- DS benchmark, which is modified

with semi-structured and unstructured data components added. The semi-structured part captures registered and guest user clicks on the retailer's website. The unstructured data captures product reviews submitted online. The workload is designed around a set of queries against the data model. It covers different categories of big data analytics proposed by McKinsey, and spans three different dimensions including data sources, query processing types and analytic techniques. The authors reported results of their experiments on Aster Data database using SQL-MR.

Some benchmarks, such as GRAPH 500 [23] and LinkBench [24] are only for graph database evaluation. There are only two kernels in the GRAPH 500 benchmark, i.e. a kernel to construct the graph from the input tuple list, and an additional computational kernel to perform breadth-first search on the graph. It is far from a general benchmark for graph database. LinkBench is released by FaceBook for social graph database benchmarking. However LinkBench is a graph-serving benchmark, not a graph-processing benchmark. The difference is that the former simulates the transactional workloads of an interactive social network service, while the latter simulates an analytics workload. In our opinion, ingredients of these benchmarks, especially graph analytic workloads should be incorporated into the big data benchmark to evaluate graph data processing component in the whole system.

2.2 Conclusion of the Survey

After above reviewing of recent works on big data benchmarking, the authors would like to put all of these works into a table for readers' easy grasp.

Table 1. Comparison of existing works of big data benchmark (* for reference)

work	target	characteristics	comment
*TPC-C	RDBMS	transaction processing, simple query and update	OLTP
*TPC-H	RDBMS, Hadoop Hive	reporting, decision	OLAP
*TPC-W	RDBMS, noSQL	Web applications	Web OLTP
*SSB	RDBMS, Hadoop Hive	reporting, decision	OLAP
*TPC-DS	RDBMS, Hadoop Hive	reporting query, ad hoc query, iterative query, data mining query	OLAP
TeraSort	RDBMS, Hadoop	data sorting	sorting only
YCSB	noSQL database	cloud based data serving	Web OLTP
REF 11	unstructured data management system	unstructured data only edge detection, proximity search, data scanning, data fusion	not representative enough
GRAPH 500	graph noSQL database	graph data processing only	not representative enough
LinkBench	RDBMS, graph noSQL database	modeling Facebook real life application graph data processing only	not representative enough
DFSIO	Hadoop	file system level benchmark	not representative enough
Hive performance benchmark	Hadoop Hive	GREP, selection, aggregation, join and UDF aggregation only	not representative enough
GridMix	Hadoop	Mix of Hadoop jobs	not representative enough
PUMA	MapReduce	term-vector, inverted-index, self-join, adjacency-list, k-means, classification, histogram-movies, histogram-ratings, sequence-count, ranked inverted index, Tera-sort, GREP, word-count	comprehensive workload

Table 1. (*continued*)

MRBench	MapReduce	TPC-H queries	OLAP
HiBench	MapReduce	micro benchmarks (sort, word count and TeraSort) Web search (Nutch Indexing and page rank) machine learning (Bayesian classification and K-means clustering) HDFS benchmark (file system level benchmark)	comprehensive workload
CloudRank-D	RDBMS, Hadoop	basic operations for data analysis, classification, clustering, recommendation, sequence learning, association rule mining, and data warehouse queries	comprehensive workload
BigBench	RDBMS, Hadoop	covers data models of structured, semi-structured and unstructured data addresses variety, velocity and volume aspects of big data systems	comprehensive workload

3 Additional Considerations

3.1 Representative Applications, Component or End-to-End Benchmark

To make the big data benchmark useful for evaluation target systems by customer, and help vendors and researchers to continuously improve their technologies, the workloads of the big data benchmark should be derived from real life applications.

One of possible application prototypes would be internet-scale applications and data management systems of large internet companies such as Facebook, Google etc. There is a gap between privacy protection and willingness to be open when building a benchmark form real applications from these companies.

Can a prototype represent thousands of applications in the big data field? A prototype could be built using a combination of a bottom-up method and a top-down method, which integrates characteristics of real life applications, and abstracts some basic operations from them in the mean time.

Since only one tool can not fit all user requirements, there should be more than one component in a big data processing platform. In a big data system, various data is used in combination to extract useful information for decision making. The data is usually *multi-structured*. Following components are needed (according to real life application requirements, these components could be used by combination), (1) *streaming computing engine*, which is for stream data processing, (2) *structured data processing engine*, which could be built upon traditional OLAP-purpose RDBMS, (3) and *unstructured data processing engine*, which could be based on Hadoop and some noSQL databases. In certain applications graph analytics are indispensable.

In such a setting, both component based benchmarking and end-to-end benchmarking are required. Customers can select a single benchmark from the benchmark suite to evaluate some specific tool of the platform, such as a stream computing engine or a graph database.

On the other hand, the big data analysis is basically a workflow of analytic tasks instead of a single analytic task. These analytical processes can run across multiple tools in a big data platform. For example [25], social network interaction data can be loaded into Hadoop, sentiment data and social 'Handles' are extracted by text mining techniques. Then the sentiment is connected to a customer and scored, to add sentiment scores into the data warehouse. Analysis is done to identify unhappy

customers and some actions could be taken to retain them. Moreover, the extracted social handles can be loaded into a graph database for further social network analysis to determine important relationships for possible cross-selling. So end-to-end benchmarking is also needed for cross-tools analytic.

3.2 Workloads – From Simple SQL Queries to Complex Analytics

The workloads should be carefully considered. On the whole, the workloads for a big data benchmark should not only contain simple SQL queries, machine learning, data mining, as well as information retrieval algorithms should also be included.

In our opinion, graph analytics becomes more and more important. It should be an indispensable part of the workloads. In some application scenarios such as multi media data processing and scientific data processing, some image manipulation algorithms and array/matrix algorithms should also be considered.

The workloads could be organized by a two-level of aggregations as depicted in figure 1. Various workflows could be composed using the basic analytic algorithms and SQL queries. Upon the basis, several typical big data application scenarios are built by combining the workflows. Customers could use some or all of the application scenarios to test the target system for their business objectives.

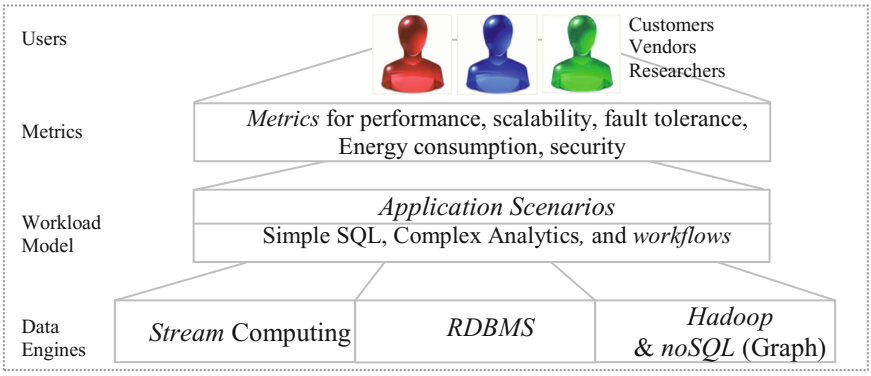


Fig. 1. Building blocks of a big data benchmark

Note: graph storage and processing is an indispensable part of a big data platform.

3.3 Not Only Performance Metrics, But Also Other Metrics

Traditional benchmarks focus on performance metrics of target systems. For big data benchmarking that is not enough. More metrics should be considered, including *scalability, fault tolerance, energy saving, and security guarantee*.

To tackle the problem of big data, the processing software needs to run on a large cluster, mostly the cluster is deployed on virtual nodes in a cloud environment. The scalability of the big data system should be tested. A cloud platform is elastic, not only it can scale out, but also it can scale in. The dynamic scalability asks people to test the target big data system when the size of underlying cluster dynamically changes according to cost and performance requirements.

Fault tolerance is also an important factor when failures of nodes are common. In benchmarking, some failures should be purposely injected into the target system to see whether the big data processing platform can handle the failures to meet the service level requirement.

In the big data era, using large number of machines to processing the big data may consume huge energy. Energy consumption is a critical metric that both vendors and researchers should take serious. Lastly security is also an important issue, some work should be done to incorporate security test into the benchmark framework.

4 Conclusions

The paper gives a review of recent works in big data benchmarking. Industry and academia are on the road toward a comprehensive big data benchmark suit. Big data benchmarking is different from traditional transaction processing benchmarking and analytic processing benchmarking in several aspects, such as data volume, data variety etc. Some considerations of us are provided for the historical process, including: component based benchmarking should be used together with application-oriented end-to-end benchmarking, complex data mining and machine learning algorithms should be run against the system beyond simple SQL queries to test deep analytic capability of the target system, and other critical metrics such as energy consumption besides response time and throughput should also be measured.

Acknowledgements. This work is funded by the NSF of China under Grants No. 61170013, NSF of Jiangsu Province under Grant No. BK2012578.

References

1. Ventana Research: Hadoop and Information Management: Benchmarking the Challenge of Enormous Volumes of Data (2013), http://www.ventanaresearch.com/research/benchmarkDetail.aspx?id=1663
2. Big Data Top 100: An open, community-based effort for benchmarking big data systems (2013), http://bigdatatop100.org/benchmarks
3. Hemsoth, N.: A New Benchmark for Big Data (2013), http://www.datanami.com/datanami/2013-03-06/a_new_benchmark_for_big_data.html
4. Kim, K., Jeon, K., Han, H., Kim, S.G., Jung, H., Yeom, H.Y.: MRBench: A Benchmark for MapReduce Framework. In: Proceedings of ICPADS, pp. 11–18. IEEE Press, Melbourne (2008)
5. Loebman, S., Nunley, D., Kwon, Y., Howe, B., Balazinska, M., Gardner, J.P.: Analyzing massive astrophysical datasets: Can Pig/Hadoop or a relational DBMS help? In: Proceedings of CLUSTER, pp. 1–10. IEEE Press, New Orleans (2009)
6. Moussa, R.: TPC-H Benchmark Analytics Scenarios and Performances on Hadoop Data Clouds. In: Benlamri, R. (ed.) NDT 2012, Part I. CCIS, vol. 293, pp. 220–234. Springer, Heidelberg (2012)
7. Ahmad, F., Lee, S., Thottethodi, M., Vijaykumar, T.N.: PUMA: Purdue MapReduce Benchmarks Suite. Purdue Technical Report TR-ECE-12-11 (2012)

8. Chen, Y., Alspaugh, S., Ganapathi, A., Griffith, R., Katz, R.: SWIM - Statistical Workload Injector for MapReduce (2013), https://github.com/SWIMProjectUCB/SWIM/wiki
9. Chen, Y.P., Raab, F., Katz, R.H.: From TPC-C to Big Data Benchmarks: A Functional Workload Model. UC Berkeley Technical Report UCB/EECS-2012-174 (2012)
10. Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R.: Benchmarking Cloud Serving Systems with YCSB. In: SoCC, pp. 143–154. ACM Press, Indianapolis (2010)
11. Smullen, C.W., Shahrukh, I.V., Tarapore, R., Gurumurthi, S.: A Benchmark Suite for Unstructured Data Processing. In: Proceedings of the International Workshop on Storage Network Architecture and Parallel I/Os (in Conjunction with MSST), pp. 79–83. IEEE Press, San Diego (2007)
12. Huang, S.S., Huang, J., Dai, J.Q., Xie, T., Huang, B.: The HiBench Benchmark Suite: Characterization of the MapReduce-Based Data Analysis. In: ICDE Workshops on Information & Software as Services, pp. 41–51. IEEE Press, Long Beach (2010)
13. Nyberg, C., Shah, M.: Sort benchmark (2012), http://sortbenchmark.org/
14. TeraSort. TeraSort Benchmark (2012), http://sortbenchmark.org/
15. Malley, O.O., Murthy, A.C.: Winning a 60 Second Dash with a Yellow Elephant (2009), http://sortbenchmark.org/Yahoo2009.pdf
16. GridMix. GridMix Benchmark (2012), http://hadoop.apache.org/docs/r1.1.1/gridmix.html
17. Jia, Y., Shao, Z.: A Benchmark for Hive, PIG and Hadoop (2012), http://issues.apache.org/jira/browse/HIVE-396
18. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., Stonebraker, M.: A comparison of approaches to large-scale data analysis. In: SIGMOD, pp. 165–178. ACM Press, Rhode Island (2009)
19. DFSIO program. DFSIO of Hadoop source distribution (2012), src/test/org/apache/hadoop/fs/TestDFSIO
20. Luo, C.J., Zhan, J.F., Jia, Z., Wang, L., Lu, G., Zhang, L.X., Xu, C.Z., Sun, N.H.: CloudRank-D: benchmarking and ranking cloud computing systems for data processing applications. Frontier of Computer Science 6(4), 347–362 (2012)
21. UCSD Center for Large Scale Data Systems Research: Big Data Benchmarking Workshops (2013), http://clds.ucsd.edu/bdbc/workshops
22. Ghazal, A., Rabl, T., Hu, M., Raab, F., Poess, M., Crolotte, A., Jacobsen, H.A.: BigBench: Towards an Industry Standard Benchmark for Big Data Analytics. In: SIGMOD. ACM Press, New York (2013)
23. Graph 500: Graph 500 Benchmark 1 (2013), http://www.graph500.org/specifications
24. King, R.: Facebook releasing new Social Graph database benchmark: LinkBench (2013), http://www.zdnet.com/facebook-releasing-new-social-graph-database-benchmark-linkbench-7000013356/
25. Ferguson, M.: Architecting a Big Data Platform for Analytics. A Whitepaper Prepared for IBM (2012)

Spectral Clustering Algorithm Based on Local Sparse Representation

Sen Wu, Min Quan, and Xiaodong Feng

Dongling School of Economics and Management,
University of Science and Technology Beijing
Beijing, 100083, P.R. China
wusen@manage.ustb.edu.cn,
quanmin900112@hotmail.com,
fengxd1988@gmail.com

Abstract. Clustering based on sparse representation is an important technique in machine learning and data mining fields. However, it is time-consuming because it constructs l_1-graph by solving l_1-minimization with all other samples as dictionary for each sample. This paper is focused on improving the efficiency of clustering based on sparse representation. Specifically, the Spectral Clustering Algorithm Based on Local Sparse Representation (SCAL) is proposed. For a given sample the algorithm solves l_1-minimization with the local k nearest neighborhood as dictionary, constructs the similarity matrix by calculating sparsity induced similarity (SIS) of the sparse coefficients solution, and then uses spectral clustering with the similarity matrix to cluster the samples. Experiments using face recognition data sets ORL and Extended Yale B demonstrate that the proposed SCAL can get better clustering performance and less time consumption.

Keywords: Spectral Clustering, Weight Matrix, Sparse Representation, $k - nn$.

1 Introduction

Spectral clustering algorithm, as one kind of important clustering algorithms, searches for clusters in the full feature space, and it is equivalent to graph min-cut problem based on a graph structure constructed from the original samples in vector space. It has been increasingly and widely developed and applied in recent years, such as financial time series [1], cluster analysis of spam images[2], cancer diagnosis and treatment[3]. Compared with the traditional clustering methods, spectral clustering has the advantage to cluster the arbitrary shape of samples space and finds the global optimal solution [4].

In recent years, sparse representation, the algorithmic problem of computing sparse linear representation with respect to an over-complete dictionary of base elements, has attracted a great attention in machine learning and pattern recognition [5]. Semi-supervised learning by sparse representation [6] assumes

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 628–635, 2013.

that each sample can be reconstructed by the sparse linear combination of other samples and constructs a l_1 graph by solving l_1 minimization problems.

The basic idea of unsupervised spectral clustering based on sparse representation (SCSR)[7] solves the sparse decomposition of each sample in the form of the linear combination of other samples; constructs the weight matrix between data samples using the sparse coefficients solution; then the weight matrix of this graph is used as the weight matrix for spectral clustering to get the cluster result. In SCSR, finding sparse representation is fast in theory if the sparsest solution is found(i.e. few non-zero coefficients)[5]. In practice, however, when the number of samples is large, the computational time cost of solving the sparse representation from all other instances is too expensive.

To overcome the time consumption deficiency of finding sparse representation, we propose to solve the sparse representation problem in a local domain to obtain an approximate solution and then to construct a local graph for spectral clustering, called spectral clustering algorithm based on local sparse representation (SCAL). Experiments conducted on real datasets demonstrated that the proposed SCAL algorithm can reduce the computational complexity.

The rest of the paper is organized as follows: in Section 2, we review an related work of spectral clustering based on sparse representation, followed by our proposed algorithm in Section 3. Section 4 demonstrates the detailed experiment results, and our work is concluded in Section 5.

2 Spectral Clustering Based on Sparse Representation

Sparse representation means to represent a sample as a linear combination of a few atoms of a given dictionary. Mathematically, given sample set X= $[x_1, x_2, ..., x_N] \in R^{M \times N}$,where N is the total number of samples and M is the number of fatures of each sample; a dictionary $D \in R^{M \times K}$, where K is the numeber of atoms in D, the sparse representation problem can be stated as :

$$\hat{\alpha} = \arg\min_{\alpha} ||\alpha||_0 \quad s.t. x_i = D\alpha \tag{1}$$

Where $||\alpha||_0$ is the l_0 pseudo-norm of the coefficient vector $\alpha \in R^K$,which counts the number of nonzero entries in a vector. However, finding the sparsest solution of (1) is NP-hard. So Wright et al. [8] proposed to solve the following l_1-minimization problem instead:

$$\hat{\alpha} = \arg\min_{\alpha} ||\alpha||_1 \quad s.t. x_i = D\alpha \tag{2}$$

Where $||\alpha||_1$ is the l_1 norm of the coefficient vector α, which is simply the sum of the absolute values of the columns. In the noisy case the equality constraint must be relaxed as well. An alternative then is to solve the unconstrained problem,

$$\hat{\alpha} = \arg\min_{\alpha} ||x_i - D\alpha||_2^2 + \lambda||\alpha||_1 \tag{3}$$

Where λ is a parameter that balances the tradeoff between reconstruction error and sparsity.

Spectral Clustering Based on Sparse Representation is depended on a basic assumption that any data sample in the data set can be well represented by the linear combination of a small number samples from the same cluster. Nevertheless, for each sample, unsupervised clustering does not have any priori information which samples are in the same cluster, To successfully identify the potential small set to reconstruct each sample,John Wright and Yi Ma etc. [9] used equation (2) to represent each sample in form of the linear combination of other samples in the dataset. For each sample x_i, $i\in$ 1,2,...,N, set X_i=X\x_i=[x_1,x_2,...,x_{i-1}, x_{i+1},...,x_N], then the reconstruction weight α_i for x_i can be calculated by solving the following l_1-minimization problem:

$$\hat{\alpha_i} = \arg\min_{\alpha_i} ||x_i - X_i\alpha_i||_2^2 \ + \lambda||\alpha_i||_1 \tag{4}$$

Theoretically, the computational complexity to obtain the sparse representation is approximately to $O(t^2$(N-1)) for each sample, where t is the number of non-zero entries in reconstruction coefficients and N-1 is the number of samples in dictionary X_i. In practice, due to there are many very small non-zero reconstruction coefficients in the obtained solution, the computation complexity to find the sparse reconstruction coefficients tend to be about $O(N^3)$ for each sample as t tends to be N.

According to the coefficient vector α, a sparse weight graph G can be constructed using different methods to characterize the relationship between samples. The commonly used weight matrix construction approaches are as follows.

Direct Construction[9]: $W_{ij} = |\alpha_j^i|$, if $i > j$, and W_{ij}=$|\alpha_{j-1}^i|$ if $i < j$,where α_j^i is the coefficient corresponding to the jth sample basis function in representation of sample x_i.

Symmetric Weight (CSR)[10]: on the basis of Direct Construction, if $\alpha_j^i \neq 0$, set the weight W_{ij}=$|\alpha_j^i|$, $1 \leq i, j \leq N$, and then W=$(W^T + W)/2$.

Sparsity Induced Similarity (SIS)[11]: The similarity between x_i and x_j, $1 \leq i, j \leq N$, $i \neq j$ is defined as

$$W_{ij} = \frac{\max\{\alpha_i^j, 0\}}{\sum\limits_{k=1,k\neq i}^{N} \max\{\alpha_i^k, 0\}}$$

This matrix is not necessarily symmetric, to ensure symmetric, the final similarity between x_i and x_j is defined as $W_{ij} = (W_{ij} + W_{ji})/2$, and set $W_{ii} = 0$.

In the concrete realization of the process, the similarity matrix W is constructed differently in different algorithms. When we get the weight matrix W, we can use the basic framework of spectral clustering[12] to cluster the dataset into K clusters.

3 The Proposed Algorithm: SCAL

In this section, we propose to solve the sparse decomposition problem in a local domain to obtain an approximate solution, this method solves the same problem

in the local neighborhood of each sample to improve the efficiency. Furthermore, in order to better reflect the relationship between samples, we use Sparsity Induced Similarity (SIS) of sparse coefficients to induce the weights of the directed l_1-graph.

Firstly, *k-nn* rules is used to find the k neighbors for each sample x_i from X_i=$X\backslash x_i$=[x_1,x_2,...,x_{i-1}, x_{i+1},...,x_N] , and defining $N\_id \in R^{N\times k}$ as the location index of the nearest k neighbors in the original data X. Notice that the k nearest neighbors are measured by the traditional Euclidean distance.

Then, l_1-norm minimization is bulit to slove the sparse representation problem in a local domain to obtain sparse coefficients for each sample x_i

$$\hat{\alpha}_i = \arg \min_{\alpha_i} ||X_{k(x_i)}\alpha_i - x_i||_2^2 + \lambda||\alpha_i||_1 \tag{5}$$

$X_{k(x_i)}$ denotes the data matrix of the k nearest neighbors of x_i. So, we can get sparse coefficients $\alpha_i \in R^k$ as the best reconstruction coefficients to represent the sample x_i. Hence the coefficient matrix is noted as A=[$\alpha_1, \ldots, \alpha_N$] $\in R^{k\times N}$.

Due to the size of coefficient matrix A is $k\times N$, it should be transformed to a $N\times N$ matrix W before calculating the similarity of two samples. The transformation function is described as follows:

$$W_{ij} = \begin{cases} A_{ri} \;\; if \; j = N\_id_{ir} \;\; i, \; r \in \{1, \ldots, k\} \\ 0 \;\; otherwise \end{cases} \tag{6}$$

Then we use Sparsity Induced Similarity (SIS) to define the similarity between x_i and x_j:

$$S_{ij} = \begin{cases} \frac{\max\{W_{ij},0\}}{\sum_{j=1}^{N} \max\{W_{ij},0\}} & if \; i \neq j \\ 0 \quad otherwise \end{cases} \tag{7}$$

Note that this matrix is not necessarily symmetric, it can be converted to be a symmetric matrix , that is: $S_{ij} = (S_{ij} + S_{ji})/2$.

After getting the weight matrix S, it can be seen as the input weight matrix of the basic framework of spectral clustering to group the sample sets X into K clusters. When the number of local neighborhood k is set N-1, our proposed method is the equal to SCSR. In manifold learning the local linearity is used to capture the local geometric structure[13] and each data point on the manifold can be locally approximated by a linear combination of its nearby points[14]. Especially the local nonnegative linear reconstructing coefficients are used to discover the natural class structure[15]. So the advocated algorithm in this paper is at least theoretically possible.

In our algorithm, the system of liner equations in (5) is over-determined. The computational complexity to obtain the sparse representation is about $O(t^2k)$ for each sample, where k is the number of the neighbors of the sample, t is the number of non-zero entries in reconstruction coefficients, and $t \leq k$. Compared with the original spectral clustering based on sparse representation, the computational cost will be saved remarkably when $k \ll N$.

4 Experiments

We conduct experiments on face recognition data sets ORL and Extended Yale B to evaluate the effectiveness of the proposed algorithm SCAL. 400 and 600 images are randomly selected from ORL data set and Extended Yale B data set respectively where each class contains the same number of images, and each image is manually cropped and normalized to the size of 32-by-32 pixels in our experiments. The cluster number (K) of two data sets are 40 and 10 respectively. Two spectral clustering algorithms based on sparse representation with all other samples as dictionary in [10](denoted as CSR) and [11] (denoted as SIS) are compared with the proposed SCAL. The difference of CSR and SIS is the different similarity construction described in the above section.

In our experiment, we use the approach and open source tool in [16] to solve the l_1-norm constraint least square minimization problem[1] in Equation(5), since it is a specialized interior-point method for solving large scale problem. The regularization parameter λ and the relative target duality gap tolerance ε for the two data sets is set with the highest clustering accuracy[17], described in Table 1.

Table 1. The regularization parameter λ and the relative target duality gap tolerance ε in 2 data sets respectively

parameters	ORL	Extended Yale B
λ	0.05	0.001
ε	1	0.005

4.1 Clustering Accuracy

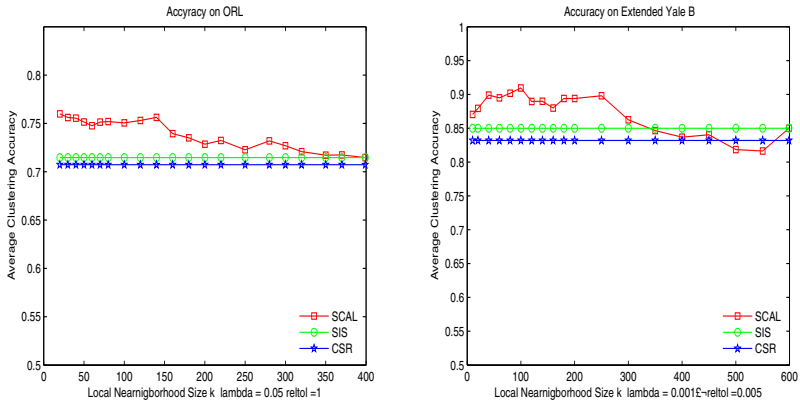
In this subsection, we adopt an external criterion which measures the degree of correspondence between the clusters obtained from our clustering algorithms and the true classes. The clustering accuracy r [17] is defined as

$$r = \frac{\sum_{i=1}^{K} a_i}{N}$$

Where a_i is the number of instances occurring in both cluster i and its corresponding class, K is the cluster number and N is the number of instances in the data set.

In ORL and Extended Yale B, all the samples are resized into 32×32 pixels. For two datasets we computed the clustering accuracy with different local neighbor number as following figures. Each step in the K-means of spectral clustering is repeated for fifty times to reduce the random influence, and the average clustering accuracy results are given in Fig.1.

[1] We use the solver for l_1 constraint minimization l_1ls.m which is downloaded from (`http://www.stanford.edu/~boyd/l1_ls/`).



(a) `ORL data set` (b) `Extended Yale B data set`

Fig. 1. Clustering accuracy on ORL data set and Extended Yale B data set

As can be seen from these figures, when the number of local neighbors is much small, the clustering accuracy of SCAL is evidently higher than other two algorithms. It's because local neighbors with appropriate number can well reflect the local geometric structure of data points. When the number of local neighbors become bigger, the clustering accuracy of SCAL approximates to that of SIS[11] using sparsity induced similarity to character the similarity of two data points. Moreover, it can get the best clustering accuracy in SCAL under small number of local neighbors. The clustering accuracy and the corresponding number of local neighbors are represented in Table 2.

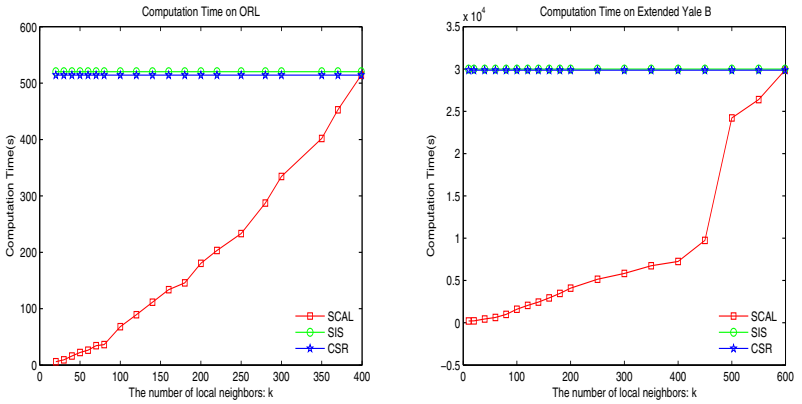
Table 2. The clustering accuracy and the corresponding number k of local neighbors

Data Sets	Symmetric Weight	SIS	Local Sparse Representation	k
ORL	0.7073	0.7147	**0.7599**	20
Extended Yale B	0.8320	0.8500	**0.9096**	100

4.2 Computation Time on Two Datasets

Considering the same condition described in Subsection 4.1, we record the computational time of SCAL and compared CSR and SIS to solve the sparse representation problem on two data sets. The computational times are presented in Fig.2.

As shown in fig.2., the smaller the number of local neighbors is, the faster it is to resolve the sparse representation. The results validate that the computation time of SCAL can approximate to the CSR and SIS when the number of local neighbors increases.



(a) `ORL data set` (b) `Extended Yale B data set`

Fig. 2. The computation time on ORL data set and Extended Yale B data set

5 Concluding Remarks

In this paper we propose spectral clustering algorithm based on local sparse representation (SCAL) to solve the problem of high computational complexity to find sparse representation in SCSR. In SCAL, the sparse representation is resolved in the local neighborhood of each sample, and the similarity matrix is constructed by calculating sparsity induced similarity of the sparse coefficients solution, using spectral clustering with this matrix as input weight matrix. Experiments on face recognition data sets ORL and Extended Yale B demonstrate that the proposed algorithm can reduce the computation complexity and the clustering accuracy is higher when the number of local neighbors is small.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No.71271027, by the Fundamental Research Funds for the Central Universities of China under Grant No. FRF-TP-10-006B and by the Research Fund for the Doctoral Program of Higher Education under Grant No. 20120006110037.

References

1. Warren Liao, T.: Clustering of time series data-a survey. Pattern Recognition 38, 1857–1874 (2005)
2. Gao, Y., Choudhary, A., Hua, G.: A nonnegative sparsity induced similarity measure with application to cluster analysis of spam images. In: 2010 IEEE International Conference on Acoustics Speech and Signal Pro-cessing (ICASSP), pp. 5594–5597. IEEE (2010)
3. Yu, Z., Li, L., You, J., Wong, H., Han, G.: SC3: Triple Spectral Clustering-Based Consensus Clustering Framework for Class Discovery from Cancer Gene Expression Profiles. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 9, 1751–1765 (2012)

4. Wang, H., Chen, J., Huang, S., Guo, K.: A Heuristic Initialization-Independent Spectral Clustering. In: 2010 Fifth International Conference on Internet Computing for Science and Engineering (ICICSE), pp. 81–84. IEEE (2010)
5. Li, C., Guo, J., Zhang, H.: Local sparse representation based classification. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 649–652. IEEE (2010)
6. Yan, S., Wang, H.: Semi-supervised Learning by Sparse Representation. In: 2009 SIAM Conference on Data Mining, pp. 792–801 (2009)
7. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 2790–2797. IEEE (2009)
8. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 210–227 (2009)
9. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse representation for computer vision and pattern recognition. P IEEE 98, 1031–1044 (2010)
10. Jiao, J., Mo, X., Shen, C.: Image clustering via sparse representation. In: Boll, S., Tian, Q., Zhang, L., Zhang, Z., Chen, Y.-P.P. (eds.) MMM 2010. LNCS, vol. 5916, pp. 761–766. Springer, Heidelberg (2010)
11. Cheng, H., Liu, Z., Yang, J.: Sparsity induced similarity measure for label propagation. In: 2009 IEEE 12th International Conference on Image clustering via sparse representation, pp. 317–324. IEEE (2009)
12. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 2, pp. 849–856 (2002)
13. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
14. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems, pp. 2223–2231 (2009)
15. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering 20, 55–67 (2008)
16. Kim, S.J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: A method for large-scale l1-regularized least squares. IEEE Journal on Selected Topics in Signal Processing 4, 606–617 (2007)
17. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Disc. 2, 283–304 (1998)

Author Index