

Xizhao Wang
Witold Pedrycz
Patrick Chan
Qiang He (Eds.)

Communications in Computer and Information Science

481

Machine Learning and Cybernetics

13th International Conference
Lanzhou, China, July 13–16, 2014
Proceedings

 Springer

Editorial Board

Simone Diniz Junqueira Barbosa

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rio de Janeiro, Brazil*

Phoebe Chen

La Trobe University, Melbourne, Australia

Alfredo Cuzzocrea

ICAR-CNR and University of Calabria, Cosenza, Italy

Xiaoyong Du

Renmin University of China, Beijing, China

Joaquim Filipe

Polytechnic Institute of Setúbal, Setúbal, Portugal

Orhun Kara

TÜBİTAK BİLGEM and Middle East Technical University, Ankara, Turkey

Igor Kotenko

*St. Petersburg Institute for Informatics and Automation of the Russian
Academy of Sciences, St. Petersburg, Russia*

Krishna M. Sivalingam

Indian Institute of Technology Madras, Chennai, India

Dominik Ślęzak

University of Warsaw and Infobright, Warsaw, Poland

Takashi Washio

Osaka University, Osaka, Japan

Xiaokang Yang

Shanghai Jiao Tong University, Shanghai, China

More information about this series at <http://www.springer.com/series/7899>

Xizhao Wang · Witold Pedrycz
Patrick Chan · Qiang He (Eds.)

Machine Learning and Cybernetics

13th International Conference
Lanzhou, China, July 13–16, 2014
Proceedings

Editors

Xizhao Wang
Hebei University
Baoding
China

Patrick Chan
South China University of Technology
Guangzhou
China

Witold Pedrycz
Department of Electrical and Computer Eng.
University of Alberta
Edmonton
Alberta
Canada

Qiang He
Hebei University
Baoding
China

ISSN 1865-0929

Communications in Computer and Information Science

ISBN 978-3-662-45651-4

DOI 10.1007/978-3-662-45652-1

ISSN 1865-0937 (electronic)

ISBN 978-3-662-45652-1 (eBook)

Library of Congress Control Number: 2014956933

Springer Heidelberg New York Dordrecht London

© Springer-Verlag Berlin Heidelberg 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media
(www.springer.com)

Preface

A warm welcome to you all to our 13th International Conference on Machine Learning and Cybernetics (ICMLC 2014) in Lanzhou. Since its inception in 2002, ICMLC has established itself as a leading conference for the rendezvous of machine learning and cybernetics, an important annual event for researchers and practitioners to gather together to share their important findings, exchange views, and explore collaboration opportunities. This year apart from the keynote speeches and paper presentations that help to keep us informed of the latest developments, we have prepared for the attendees various sessions and activities that cater to the needs of our up-and-coming researchers. These include free tutorials and the Lotfi Zadeh Best Paper Award. We hope you will also take advantage of the various social activities which have been organized for you to extend your network of research collaboration around the world.

The conference began with an opening ceremony and the conference program featured a welcome speech, two tutorials, two keynote speeches, and Panel Discussion presentations by local and international experts. During the three-day program, all paper presentations were given in four parallel sessions. The conference ended with a closing ceremony. The conference received more than 421 papers, each paper was carefully reviewed by the Program Committee members, and finally 45 papers were selected for this CCIS volume.

All the papers selected for this volume underwent English editing. They are classified into ten categories, i.e., classification and semi-supervised learning, clustering and kernel, application to recognition, sampling and big data, application to detection, decision tree learning, learning and adaptation, similarity and decision making, learning with uncertainty and improved learning algorithms and applications. The authors of these papers are from 11 different countries or areas. These papers may be of interest to those working in the field of machine learning and its applications.

The success of this conference would not have been possible without the great efforts of our Program Chairs, Program Committee members, reviewers, and our supporting colleagues working behind the scene. Certainly our conference could not succeed without your invaluable support. We hope you will find this conference a valuable and useful experience and your stay in Lanzhou enjoyable and memorable. We look forward to your continuing support and participation in our conferences to come.

September 2014

Xizhao Wang
Witold Pedrycz
Patrick Chan
Qiang He

Conference Organization

The 13th International Conference on Machine Learning and Cybernetics, ICMLC 2014, was organized by Key laboratory of Machine Learning and Computational Intelligence in Hebei Province, and College of Mathematics and Computer Science, Hebei University, Hebei, China.

Honorary Conference Chairs

Hongrui Wang	President Hebei University, China
Michael Smith	IEEE Systems, Man and Cybernetics Society, USA
William A. Gruver	Simon Fraser University, Canada

General Chairs

Daniel S. Yeung	South China University of Technology, China
Xizhao Wang	Hebei University, China

Program Chairs

Shyi-Ming Chen	National Taiwan University of Science and Technology, Taiwan
Qinghua Hu	Tianjin University, China
Honghai Liu	University of Portsmouth, UK
Ching-Chih Tsai	National Chung Hsing University, Taiwan

Publication Chair

Patrick Chan	South China University of Technology, China
--------------	---

Treasurer

Eric Tsang	Macau University of Science and Technology, Macau
------------	--

Local Arrangement Chair

Zeng-Tai Gong	Northwest Normal University, China
Robert P. Woon	IEEE Systems, Man, and Cybernetics Society, USA

Conference Secretaries

Patrick Chan	South China University of Technology, China
Wing Ng	South China University of Technology, China

Technical Program Committee

Yi Cai	South China University of Technology, China
Junyi Chai	The Hong Kong Polytechnic University, Hong Kong
Patrick Chan	South China University of Technology, China
Chi-Yuan Chang	Jinwen University of Science and Technology, Taiwan
Degang Chen	North China Electric Power University, China
Duan-Yu Chen	Yuan Ze University, Taiwan
Junfen Chen	Universiti Sains Malaysia, Malaysia
Shyi-Ming Chen	National Taiwan University of Science and Technology, Taiwan
Shou-Hsiung Cheng	Chienkuo Technology University, Taiwan
Been-Chian Chien	National University of Tainan, Taiwan
Shih-Hao Fang	Yuan Ze University, Taiwan
William Gruver	Simon Fraser University, Canada
Fei Guan	Beijing University of Science and Technology, China
Qiang He	Hebei University, China
Yulin He	The Hong Kong Polytechnic University, Hong Kong
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Cheng-Hsiung Hsieh	Chaoyung University of Technology, Taiwan
Qinghua Hu	Tianjin University, China
Guo-Hsing Huang	National Chin-Yi University of Technology, Taiwan
Hsu-Chih Huang	National Ilan University, Taiwan
Kao-Shing Hwang	National Sat Yet-Sen University, Taiwan
Chih-Chin Lai	National University of Kaohsiung, Taiwan
Loi Lei Lai	State Grid, China
Jiann-Shu Lee	National University of Tainan, Taiwan
John Lee	The Hong Kong Polytechnic University, Hong Kong
Li-Wei Lee	De Lin Institute of Technology, Taiwan
Shie-Jue Lee	National Sun Yat-Sen University, Taiwan
Walter Fung Sui Leung	The Hong Kong Polytechnic University, Hong Kong
Fachao Li	Hebei University of Science and Technology, China
Tzue-Hseng Li	National Cheng Kung University, Taiwan
Jiye Liang	Shanxi University, China
Chih-Min Lin	Yuan Ze University, Taiwan
Wen-Yang Lin	National University of Kaohsiung, Taiwan
James Liu	The Hong Kong Polytechnic University, Hong Kong

Jun Liu	University of Ulster, UK
Chi-Huang Lu	Hsiuping University of Science and Technology, Taiwan
Yi-Jen Mon	Taoyuan Innovation Institute of Technology, China
Wing Ng	South China University of Technology, China
Minh Nhut Nguyen	Institute of Infocomm Research, Singapore
Chen-Sen Ouyang	I-Shou University, Taiwan
Shing-Tai Pan	National University of Kaohsiung, Taiwan
Yuhua Qian	Shanxi University, China
Ying Qu	Hebei University of Science and Technology, China
Mingwen Shao	Qingdao Technological University, China
Victor R.L. Shen	National Taipei University, Taiwan
Hornng-Lin Shieh	St. John's University, Taiwan
Huang-Chia Shih	Yuan Ze University, Taiwan
Shun-Feng Su	National Taiwan University of Science and Technology, Taiwan
Feng Tian	Bournemouth University, UK
Ching-Chih Tsai	National Chung Hsing University, Taiwan
Hsien-Leing Tsai	I-Shou University, Taiwan
Kuei-I Tsai	National Chin-Yi University of Technology, Taiwan
Eric Tsang	Macau University of Science and Technology, Macau
Changzhong Wang	Bohai University, China
Cheng-Yi Wang	National Taiwan University of Science and Technology, Taiwan
Hengyou Wang	Beijing University of Civil Engineering and Architecture, China
Hsien-Chang Wang	Chang Jung Christian University, Taiwan
Hui Wang	University of Ulster, UK
Lipo Wang	Nanyang Technological University, China
Ran Wang	City University of Hong Kong, Hong Kong
Shyue-Liang Wang	National University of Kaohsiung, Taiwan
Chih-Hung Wu	National University of Kaohsiung, Taiwan
Yi-Leh Wu	National Taiwan University of Science and Technology, Taiwan
Hong Yan	City University of Hong Kong, Hong Kong
Daniel Yeung	South China University of Technology, China
Junhai Zhai	Hebei University, China
Guoli Zhang	North China Electric Power University, China
Zan Zhang	Tianjin University, China
Shixin Zhao	Shijiazhuang Tiedao University, China

Support and Sponsors

Hebei University, China
IEEE Systems, Man and Cybernetics Society, USA

South China University of Technology, China

University of Macau, Macau

Chongqing University, China

Harbin Institute of Technology, Shenzhen Graduate School, China

University of Ulster, UK

City University of Hong Kong, Hong Kong

Department of Electrical and Electronic Engineering, University of Cagliari, Italy

Hebei University of Science and Technology, China

Table of Contents

Classification and Semi-Supervised Learning

Combining Classifiers Based on Gaussian Mixture Model Approach to Ensemble Data	3
<i>Tien Thanh Nguyen, Alan Wee-Chung Liew, Minh Toan Tran, and Mai Phuong Nguyen</i>	
Sentiment Classification of Chinese Reviews in Different Domain: A Comparative Study	13
<i>Qingqing Zhou and Chengzhi Zhang</i>	
A Computer-Aided System for Classification of Breast Tumors in Ultrasound Images via Biclustering Learning	24
<i>Qiangzhi Zhang, Huali Chang, Longzhong Liu, Anhua Li, and Qinghua Huang</i>	
An Improved Approach to Ordinal Classification	33
<i>Donghui Wang, Junhai Zhai, Hong Zhu, and Xizhao Wang</i>	
Classification Based on Lower Integral and Extreme Learning Machine	43
<i>Aixia Chen, Huimin Feng, and Zhen Guo</i>	
User Input Classification for Chinese Question Answering System	52
<i>Yongshuai Hou, Xiaolong Wang, Qingcai Chen, Man Li, and Cong Tan</i>	
Fusion of Classifiers Based on a Novel 2-Stage Model	60
<i>Tien Thanh Nguyen, Alan Wee-Chung Liew, Minh Toan Tran, Thi Thu Thuy Nguyen, and Mai Phuong Nguyen</i>	

Clustering and Kernel

Comparative Analysis of Density Estimation Based Kernel Regression	71
<i>Junying Chen and Yulin He</i>	
Thermal Power Units' Energy Consuming Speciality Analysis Based on Support Vector Regression (SVR).	80
<i>Ming Zhao, Zhengbo Yan, and Liukun Zhou</i>	
Bandwidth Selection for Nadaraya-Watson Kernel Estimator Using Cross-Validation Based on Different Penalty Functions	88
<i>Yumin Zhang</i>	

A Hough Transform-Based Biclustering Algorithm for Gene Expression Data 97
Cuong To, Tien Thanh Nguyen, and Alan Wee-Chung Liew

An Effective Biclustering Algorithm for Time-Series Gene Expression Data 107
Huixin Xu, Yun Xue, Zhihao Lu, Xiaohui Hu, Hongya Zhao, Zhengling Liao, and Tiechen Li

Multiple Orthogonal K-means Hashing 117
Ziqian Zeng, Yueming Lv, and Wing W.Y. Ng

Application to Recognition

Recognizing Bangladeshi Currency for Visually Impaired 129
Mohammad M. Rahman, Bruce Poon, M. Ashraful Amin, and Hong Yan

Face Recognition Using Genetic Algorithm 136
Qin Qing and Eric C.C. Tsang

Face Liveness Detection by Brightness Difference 144
Patrick P.K. Chan and Ying Shu

Sampling and Big Data

User Behavior Research Based on Big Data. 153
Suxiang Zhang and Suxian Zhang

Stochastic Sensitivity Oversampling Technique for Imbalanced Data 161
Tongwen Rong, Huachang Gong, and Wing W.Y. Ng

Application to Detection

A Heterogeneous Graph Model for Social Opinion Detection. 175
Xiangwen Liao, Yichao Huang, Jingjing Wei, Zhiyong Yu, and Guolong Chen

A Storm-Based Real-Time Micro-Blogging Burst Event Detection System . . . 186
Yiding Wang, Ruifeng Xu, Bin Liu, Lin Gui, and Bin Tang

A Causative Attack against Semi-supervised Learning 196
Yujiao Li and Daniel S. Yeung

Decision Tree Learning

Study and Improvement of Ordinal Decision Trees Based on Rank Entropy . . . 207
Jiankai Chen, Junhai Zhai, and Xizhao Wang

Extended Space Decision Tree	219
<i>Md. Nasim Adnan, Md. Zahidul Islam, and Paul W.H. Kwan</i>	
Monotonic Decision Tree for Interval Valued Data	231
<i>Hong Zhu, Junhai Zhai, Shanshan Wang, and Xizhao Wang</i>	
Parallel Ordinal Decision Tree Algorithm and Its Implementation in Framework of MapReduce	241
<i>Shanshan Wang, Junhai Zhai, Hong Zhu, and Xizhao Wang</i>	
Learning and Adaptation	
An Improved Iterative Closest Point Algorithm for Rigid Point Registration.	255
<i>Junfen Chen and Bahari Belaton</i>	
Approachs to Computing Maximal Consistent Block.	264
<i>Xiangrui Liu and Mingwen Shao</i>	
Missing Value Imputation for the Analysis of Incomplete Traffic Accident Data	275
<i>Rupam Deb and Alan Wee-Chung Liew</i>	
Learning Behaviour for Service Personalisation and Adaptation	287
<i>Liming Chen, Kerry Skillen, William Burns, Susan Quinn, Joseph Rafferty, Chris Nugent, Mark Donnelly, and Ivar Solheim</i>	
Extraction of Class Attributes from Online Encyclopedias	298
<i>Hongzhi Guo, Qingcai Chen, and Chunxiao Sun</i>	
Selective Ensemble of Rbfnn's Based on Improved Negative Correlation Learning	308
<i>Hongjie Xing, Lifei Liu, and Sen Li</i>	
A Two-Phase RBF-ELM Learning Algorithm	319
<i>Junhai Zhai, Wenxiang Hu, and Sufang Zhang</i>	
Similarity and Decision Making	
A Study on Decision Making by Thai Software House Companies in Choosing Computer Programming Languages.	331
<i>Vasin Chooprayoon</i>	
An Improved Method for Semantic Similarity Calculation Based on Stop-Words	339
<i>Haodi Li, Qingcai Chen, and Xiaolong Wang</i>	

Learning with Uncertainty

Sensitivity Analysis of Radial-Basis Function Neural Network due to the Errors of the I.I.D Input 351
Jie Li, Jun Li, and Ying Liu

Fuzzy If-Then Rules Classifier on Ensemble Data 362
Tien Thanh Nguyen, Alan Wee-Chung Liew, Cuong To, Xuan Cuong Pham, and Mai Phuong Nguyen

Image Segmentation Based on Graph-Cut Models and Probabilistic Graphical Models: A Comparative Study 371
Maedeh Beheshti and Alan Wee-Chung Liew

Tolerance Rough Fuzzy Approximation Operators and Their Properties 379
Yao Zhang, Junhai Zhai, and Sufang Zhang

Extreme Learning Machine for Interval-Valued Data 388
Shixin Zhao and Xizhao Wang

Credibility Estimation of Stock Comments Based on Publisher and Information Uncertainty Evaluation 400
Qiaoyun Qiu, Ruifeng Xu, Bin Liu, Lin Gui, and Yu Zhou

A Fast Algorithm to Building a Fuzzy Rough Classifier 409
Eric C.C. Tsang and Suyun Zhao

Improved Learning Algorithms and Applications

Surface Electromyography Time Series Analysis for Regaining the Intuitive Grasping Capability after Thumb Amputation 421
Chithrangi Kaushalya Kumarasinghe and D.K. Withanage

Study on Orthogonal Basis NN-Based Storage Modelling for Lake Hume of Upper Murray River, Australia. 431
Ying Li, Yan Li, and Xiaofen Wang

De Novo Gene Expression Analysis to Assess the Therapeutic and Toxic Effect of Tachyplesin I on Human Glioblastoma Cell Lines. 442
Hongya Zhao, Hong Ding, and Gang Jin

An Improved Reject on Negative Impact Defense. 452
Hongjiang Li and Patrick P.K. Chan

Author Index 461

Classification and Semi-Supervised Learning

Combining Classifiers Based on Gaussian Mixture Model Approach to Ensemble Data

Tien Thanh Nguyen^{1(✉)}, Alan Wee-Chung Liew¹, Minh Toan Tran²,
and Mai Phuong Nguyen³

¹ School of Information and Communication Technology, Griffith University, QLD, Australia
{a.liew,tienthanh.nguyen2}@griffithuni.edu.au

² School of Applied Mathematics and Informatics,
Hanoi University of Science and Technology, Hanoi, Vietnam
toan.tranminh@hust.edu.vn

³ College of Business, Massey University, Albany, New Zealand
phuongnm0590@gmail.com

Abstract. Combining multiple classifiers to achieve better performance than any single classifier is one of the most important research areas in machine learning. In this paper, we focus on combining different classifiers to form an effective ensemble system. By introducing a novel framework operated on outputs of different classifiers, our aim is to build a powerful model which is competitive to other well-known combining algorithms such as Decision Template, Multiple Response Linear Regression (MLR), SCANN and fixed combining rules. Our approach is difference from the traditional approaches in that we use Gaussian Mixture Model (GMM) to model distribution of Level1 data and to predict the label of an observation based on maximizing the posterior probability realized through Bayes model. We also apply Principle Component Analysis (PCA) to output of base classifiers to reduce its dimension of what before GMM modeling. Experiments were evaluated on 21 datasets coming from University of California Irvine (UCI) Machine Learning Repository to demonstrate the benefits of our framework compared with several benchmark algorithms.

Keywords: Gaussian mixture model (GMM) · Ensemble method · Multi-classifier system · Combining classifiers · Classifier fusion · Stacking algorithm · Principle component analysis (PCA)

1 Introduction and Recent Work

Traditionally, single learning algorithm is usually employed to solve classification problems by training a classifier on a particular training set which contains hypothesis about the relationship between feature vectors and its class labels. A natural question that arises is: can we combine multiple classification algorithms to achieve higher performance than a single algorithm? This is the idea behind a class of methods called ensemble methods. Ensemble method is a method that combines models to obtain lower error rate than using a single model. “Models” in ensemble methods could include not only the implementation of many different learning algorithms, or the

creation of larger training set for the same learning algorithm, but also generating generic classifiers in combination to improve efficiency of classification task [13].

In this paper, we build an ensemble system where the prediction framework is formed by combining outputs of different classifiers (called meta-data or Level1 data). One of the most popular ensemble methods is based on Stacking [1-3]. Output of the Stacking proces is posterior probability that each observation belongs to a class according to each classifier. The set of posterior probability of all observations is called meta-data or Level1 data.

Let us denote N to be the number of observations, while K is a number of base classifiers and M stands for the number of classes. For an observation X_i , $P_k(W_j | X_i)$ is probability that X_i belongs to class W_j given by k^{th} classifier. Level1 data of all observations, a $N \times MK$ -posterior probability matrix $\{P_k(W_j | X_i)\}$ $j = \overline{1, M}$ $k = \overline{1, K}$ $i = \overline{1, N}$ is given by:

$$\begin{bmatrix} P_1(W_1 | X_1) \dots P_1(W_M | X_1) & \dots & P_K(W_1 | X_1) \dots P_K(W_M | X_1) \\ \dots & & \dots \\ P_1(W_1 | X_N) \dots P_1(W_M | X_N) & \dots & P_K(W_1 | X_N) \dots P_K(W_M | X_N) \end{bmatrix} \quad (1)$$

Level1 data of an observation X is defined in the form:

$$Level1(X) := \begin{bmatrix} P_1(W_1 | X) & \dots & P_1(W_M | X) \\ \vdots & \ddots & \vdots \\ P_K(W_1 | X) & \dots & P_K(W_M | X) \end{bmatrix} \quad (2)$$

Based on stacking, various combining algorithms have been introduced. For example, Multiple Response Linear Regression (MLR) [3], Decision Template [2], SCANN [5] are well-known combining classifiers algorithm. Recently, Zhang and Zhou [6] used linear programming to find weight that each classifier puts wording on a particular class. Sen et al. [7] introduced a model inspired by MLR by applying hinge loss function to the combiner instead of using conventional least square loss. Stacking-based algorithms are called trainable algorithms since Level1data of training set is again exploited to discover latent knowledge during the second training process.

On the other hand, fixed rule is simple and effective combining classifiers method in practice. Kittler et al. [4] presented six rules named Sum, Product, Vote, Min, Max and Average. A benefit of fixed rules is that no training on Level1 data is required. Sum and Vote rules are popular combining strategy. An issue related to fixed rule is that we cannot know a priori what rule is appropriate for a specific data source.

In this work, we focus on GMM as a classifier for Level1 data. Li et al. [11] proposed a GMM classifier based on low dimensional feature space for hyper-spectral image classification. Liu et al. [9] showed that when dimension of data is high, the effectiveness of GMM approximation is reduced. To address this problem, the dimension of data for GMM is reduced by:

$$\underset{\text{likelihood function}}{P(\mathbf{x}|\Theta)} = P_F(\mathbf{x}|\Theta) \times P_{\bar{F}}(\mathbf{x}|\Theta) = P(\mathbf{y}|\Theta^*) \times P_{\bar{F}}(\mathbf{x}|\Theta) \quad (3)$$

where Θ is model for input data and Θ^* is another model for projected data \mathbf{y} by applied PCA method on input data. F and \bar{F} in turn are principal subspace containing principal component and its orthogonal complement, respectively.

The rest of this paper is organized as followed. Section 2 describes the proposed GMM based combining classifiers model that operates on Level1 data. Experimental results on 21 common UCI datasets [12] are presented in Section 3. Finally, conclusion and future work are given.

2 The Proposed Model

2.1 Combining Classifiers Based on GMM

In our knowledge, all GMM-based approaches are conducted on Level0 data (i.e. original data) in which they suffer from severe limitations in modeling various datasets. Attributes in Level0 data are frequently diverse in measurement unit and type. As a result, GMM may not approximate distribution of Level0 data well. Level1 data, on the other hand, can be viewed as scaled result from feature domain to probability domain. Observations belonging to the same class would have nearly equal posterior probability from the base classifier. As a result, they would be located close to each other. Besides, in some situations, Level1 data has lower dimension than Level0 data. It is well known that the higher the dimension of data is, the lower the effectiveness of GMM approximation. Hence, GMM on Level1 data is expected to have better performance.

This paper presents a classifier fusion technique that apply GMM on meta-data. The novel combining classifiers model is illustrated in Fig 1. First, Stacking is applied to the training set to generate Level1 data (1). Next, observations that belong to the same class are grouped together, and the class distribution is approximated by GMM.

For i^{th} class, we propose a prediction framework based on the Bayes model

$$\underset{\text{posterior}}{P(GMM_i | \mathbf{x})} \sim \underset{\text{likelihood}}{P(\mathbf{x} | GMM_i)} \times \underset{\text{prior}}{P(GMM_i)} \quad (4)$$

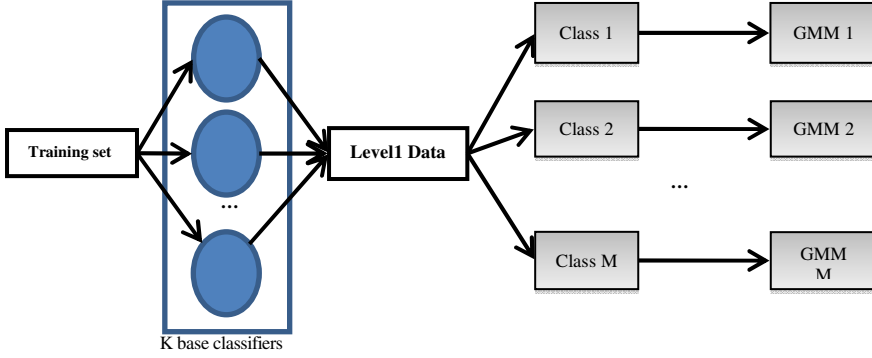


Fig. 1. GMM-based approach on Level1 data

Here likelihood function is GMM:

$$P(\mathbf{x} | GMM_i) = P(\mathbf{x} | \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}, \omega_{ip}) = \sum_{p=1}^{P_i} \omega_{ip} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}) \quad (5)$$

where

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}) = \frac{1}{(2\pi)^{MK/2} |\boldsymbol{\Sigma}_{ip}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{ip})^T \boldsymbol{\Sigma}_{ip}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{ip})\right\} \quad (6)$$

P_i is number of Gaussian components in GMM_i model, ω_{ip} , $\boldsymbol{\mu}_{ip}$, $\boldsymbol{\Sigma}_{ip}$ are mixing coefficient, mean and covariance of p^{th} component in model for i^{th} class respectively. Prior probability in (4) of i^{th} class is defined by:

$$P(GMM_i) = N_i / N \quad (7)$$

where N_i is number of observations in i^{th} class. To find parameters of GMMs, we apply Expectation Maximization (EM) algorithm by maximize the likelihood function with respect to means, covariances of components and mixing coefficients [10].

A question related to GMM is how to find the number of components. Frequently, it is user defined. Here, we propose applying Bayes Criterion Information (BIC) to find the optimal model [10]. Given a set of GMMs $\{F_j\}$ with parameters $\boldsymbol{\theta}_j$ (which are means, covariances of components and mixing coefficients of GMM), to find best model by BIC, we compute:

$$\ln P(\mathbf{x} | F_j) \approx \ln P(\mathbf{x} | F_j, \boldsymbol{\theta}_{MAP}) - \frac{1}{2} |\boldsymbol{\theta}_j| \ln N_i \quad (8)$$

where θ_{MAP} maximizes the posterior distribution and $|\theta_j|$ is number of parameters in θ_j . Note that Level1 data conveys posterior information from each classifier about the amount of supports by a classifier for an observation belonging to a class. Sometimes, there are columns in Level1 data in which $\exists k, m$ such $P_k(W_m | X_i)$ is nearly constant for all i . In this case, the covariance matrix is singular and EM is unable to solve for GMM. To overcome this problem, we randomly choose several elements in such columns and perturb their values by a small quantity before EM is applied.

Given the GMM model for each class, the class label of an observation X_{Test} is given by:

$$X_{Test} \in W_t \text{ if } t = \arg \max_{i=1, M} P(GMM_i | X_{Test}) \quad (9)$$

Algorithm: GMM for combining classifiers

Training process:

Input: Training set: L0, K base classifiers,
PiMax: maximum number of Gaussian component for
 i^{th} class.

Output: Optimum GMM for each class.

Step1: Applied Stacking to generate Level1 data.

Step2: Group Level1 data into M classes based
on class information. Compute $P(GMM_i)$ (7),
mean, and covariance for each class.

Step3: For i^{th} class

 For p=1 to PiMax

 Apply EM algorithm to find GMM with p
 components and compute BIC (8).

 End

 Select p that resulted in max(BIC) and
 associated GMM.

End

Test process:

Input: Unlabeled observation X_{Test} .

Output: Predicted label of X_{Test}

Step1: Compute Level1 data of X_{Test}

Step2: For each class

 Compute $P(X_{Test} | GMM_i)$ (5) and then posterior

$P(GMM_i | X_{Test})$

 End

Step3: Predict label of X_{Test} from (9)

2.2 GMM-PCA Model

When the number of observations is smaller than the dimension of data, GMM cannot be estimated by EM algorithm. Hence, the dimension of data needs to be reduced. We perform PCA on Level1 data and retain only the C largest eigenvalues which satisfies (10):

$$\left(\sum_{c=1}^C \lambda_c \right) \left(\sum_{c=1}^{MK} \lambda_c \right)^{-1} > 1 - \varepsilon \quad (10)$$

Then, $P(\mathbf{x}|GMM_i) \sim P(\mathbf{y}|GMM_i)$ where \mathbf{y} is the projection of \mathbf{x} on principle subspace which contains the C selected eigenvectors. Given the GMM model for each class, the class label of an observation X_{Test} is given by:

$$X_{Test} \in W_t \text{ if } t = \arg \max_{i=1,M} P(Y_{Test}|GMM_i) \times P(GMM_i) \quad (11)$$

3 Experimental Results

In our experiments, we performed 10-fold cross validation on the dataset. Moreover, to ensure objectiveness, the test was run 10 times so we had 100 error rates result for each data file. Three base classifiers, namely Linear Discriminant Analysis (LDA), Naïve Bayes and K Nearest Neighbor (with K set to 5 denoted by 5-NN) were selected. As these classifiers are different to each other in their approach, the diversity of the ensemble system is ensured. To assess statistical significance, we used paired t-test to compare two results (parameter α is set by 0.05)

In our assessment, we compared error rate of our model and five other methods: best result from the set of base classifiers, best result based on fixed combining rules, SCANN, MLR, Decision Template (with measure of similarity S_1 [2] defined as

$$S_1(Levell(X), DT_i) = \frac{\|Levell(X) \cap DT_i\|}{\|Levell(X) \cup DT_i\|} \text{ where } DT_i \text{ is Decision Template of } i^{th}$$

class and $\|\alpha\|$ is the relative cardinality of the fuzzy set α), GMM on Level0 data,. Here we used 6 fixed rules namely Sum, Product, Min, Max, Vote, Median to choose the best result based on their outcome on test set. Experimental results of 21 UCI files are showed in Table 2, 3 and 4.

In Table 2, we reported error rate of all 3 base classifiers and chose best result based on their performance on test set. We see that both GMM and GMMPCA-based approach on meta-data outperform any base classifiers. GMM posts 6 wins and only 3 losses while GMMPCA has 7 wins and 3 losses (Table 5).

GMM and GMMPCA on Level1 data are observed to perform better than GMM on Level0 data, posting up to 16 and 17 wins, respectively. Our model is only poorer than GMM on Level0 data on Ring files (2.09%). This is not surprising because the Ring dataset was drawn from multivariate Gaussian distributions [12], so GMM is the best to approximate Level0 distribution in that case. Clearly, GMM on Level0 reports higher error rates than those on Level1 data as well as Rules, Decision Template, MLR and SCANN.

Table 1. UCI data files used in our experiment (*) R: Real, C: Category, I: Integer

File name	# of attributes	Attribute type (*)	# of observations	# of classes	# of attributes on Level1
Bupa	6	C,I,R	345	2	6
Pima	6	R,I	768	2	6
Sonar	60	R	208	2	6
Heart	13	C,I,R	270	2	6
Phoneme	5	R	540	2	6
Haberman	3	I	306	2	6
Titanic	3	R,I	2201	2	6
Balance	4	C	625	3	9
Fertility	9	R	100	2	6
Wdbc	30	R	569	2	6
Australian	14	C,I,R	690	2	6
Twonorm	20	R	7400	2	6
Magic	10	R	19020	2	6
Ring	20	R	7400	2	6
Contraceptive	9	C,I	1473	3	6
Vehicle	18	I	846	4	12
Iris	4	R	150	3	9
Tae	20	C,I	151	2	6
Letter	16	I	20000	26	78
Skin&NonSkin	3	R	245057	2	6
Artificial	10	R	700	2	6

Table 2. Classifying error rate of base classifiers

File name	LDA		Naïve Bayes		5-NN		Best result from base classifiers	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3693	8.30E-03	0.4264	7.60E-03	0.3331	6.10E-03	0.3331	6.10E-03
Artificial	0.4511	1.40E-03	0.4521	1.40E-03	0.2496	2.40E-03	0.2496	2.40E-03
Pima	0.2396	2.40E-03	0.2668	2.00E-03	0.2864	2.30E-03	0.2396	2.40E-03
Sonar	0.2629	9.70E-03	0.3042	7.40E-03	0.1875	7.60E-03	0.1875	7.60E-03
Heart	0.1593	5.30E-03	0.1611	5.90E-03	0.3348	5.10E-03	0.1593	5.30E-03
Phoneme	0.2408	3.00E-04	0.2607	3.00E-04	0.1133	2.00E-04	0.1133	2.00E-04
Haberman	0.2669	4.50E-03	0.2596	4.40E-03	0.2829	3.80E-03	0.2596	4.40E-03
Titanic	0.2201	5.00E-04	0.2515	8.00E-04	0.2341	3.70E-03	0.2201	5.00E-04
Balance	0.2917	2.90E-03	0.2600	3.30E-03	0.1442	1.20E-03	0.1442	1.20E-03
Fertility	0.3460	2.01E-02	0.3770	2.08E-02	0.1550	4.50E-03	0.1550	4.50E-03
Skin&NonSkin	0.0659	2.74E-06	0.1785	6.61E-06	0.0005	1.68E-08	0.0005	1.68E-08
Wdbc	0.0397	7.00E-04	0.0587	1.20E-03	0.0666	8.00E-04	0.0397	7.00E-04
Australian	0.1416	1.55E-03	0.1297	1.71E-03	0.3457	2.11E-03	0.1297	1.71E-03
Twonorm	0.0217	3.12E-05	0.0217	3.13E-05	0.0312	3.96E-05	0.0217	3.12E-05
Magic	0.2053	6.85E-05	0.2255	7.33E-05	0.1915	4.81E-05	0.1915	4.81E-05
Ring	0.2381	2.27E-04	0.2374	2.23E-04	0.3088	1.30E-04	0.2374	2.23E-04
Tae	0.4612	1.21E-02	0.4505	1.22E-02	0.5908	1.37E-02	0.4505	1.22E-02
Contraceptive	0.4992	1.40E-03	0.5324	1.42E-03	0.4936	1.70E-03	0.4936	1.70E-03
Vehicle	0.2186	1.39E-03	0.5550	2.94E-03	0.3502	2.35E-03	0.2186	1.39E-03
Iris	0.0200	1.40E-03	0.0400	2.30E-03	0.0353	1.50E-03	0.0200	1.40E-03
Letter	0.2977	8.31E-05	0.4001	1.04E-04	0.0448	1.68E-05	0.0448	1.68E-05

Besides, our approach is competitive with best result selected from fixed rules (Table 5). The files in which we have superior performance are: Ring (11.31% vs. 21.22%), Vehicle (21.66% vs. 26.45%) and Skin&NonSkin (0.04% vs. 0.06%), while on 4 files, best result from fixed rules is better than our algorithm. Note, however, that the optimal rule for a particular data source is usually not known in advanced.

Table 3. Classifying error rate of trainable combining algorithms

File name	MLR		Best result from 6 fixed rules		SCANN		Decision Template	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3033	4.70E-03	0.2970	4.89E-03	0.3304	4.29E-03	0.3348	7.10E-03
Artificial	0.2426	2.20E-03	0.2193	2.05E-03	0.2374	2.12E-03	0.2433	1.60E-03
Pima	0.2432	2.30E-03	0.2365	2.10E-03	0.2384	2.06E-03	0.2482	2.00E-03
Sonar	0.1974	7.20E-03	0.2079	8.16E-03	0.2128	8.01E-03	0.2129	8.80E-03
Heart	0.1607	4.70E-03	0.1570	4.64E-03	0.1637	4.14E-03	0.1541	4.00E-03
Phoneme	0.1136	1.75E-04	0.1407	1.95E-04	0.1229	6.53E-04	0.1462	2.00E-04
Haberman	0.2428	3.30E-03	0.2392	2.39E-03	0.2536	1.74E-03	0.2779	5.00E-03
Titanic	0.2169	4.00E-04	0.2167	5.00E-04	0.2216	6.29E-04	0.2167	6.00E-04
Balance	0.1225	8.00E-04	0.1112	4.82E-04	X	X	0.0988	1.40E-03
Fertility	0.1250	2.28E-03	0.1270	1.97E-03	X	X	0.4520	3.41E-02
Skin&NonSkin	4.79E-04	1.97E-08	0.0006	2.13E-08	X	X	0.0332	1.64E-06
Wdbc	0.0399	7.00E-04	0.0395	5.03E-04	0.0397	5.64E-04	0.0385	5.00E-04
Australian	0.1268	1.80E-03	0.1262	1.37E-03	0.1259	1.77E-03	0.1346	1.50E-03
Twonorm	0.0217	2.24E-05	0.0216	2.82E-05	0.0216	2.39E-05	0.0221	2.62E-05
Magic	0.1875	7.76E-05	0.1905	5.72E-05	0.2002	6.14E-05	0.1927	7.82E-05
Ring	0.1700	1.69E-04	0.2122	1.62E-04	0.2150	2.44E-04	0.1894	1.78E-04
Tae	0.4652	1.24E-02	0.4435	1.70E-02	0.4428	1.34E-02	0.4643	1.21E-02
Contraceptive	0.4675	1.10E-03	0.4653	1.79E-03	0.4869	1.80E-03	0.4781	1.40E-03
Vehicle	0.2139	1.40E-03	0.2645	1.37E-03	0.2224	1.54E-03	0.2161	1.50E-03
Iris	0.0220	1.87E-03	0.0327	1.73E-03	0.0320	2.00E-03	0.0400	2.50E-03
Letter	0.0427	1.63E-05	0.0760	3.94E-05	0.063	2.42E-05	0.1133	4.91E-05

Table 4. Classifying error rate of GMM-Based approaches

File name	GMM on Level0		GMM on Level1		GMM PCA on Level1	
	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.4419	5.80E-03	0.3022	5.31E-03	0.3176	5.49E-03
Artificial	0.4507	8.00E-03	0.2374	2.40E-03	0.2329	1.66E-03
Pima	0.2466	2.40E-03	0.2432	2.60E-03	0.2158	8.70E-03
Sonar	0.3193	1.26E-02	0.2009	6.20E-03	0.1974	6.90E-03
Heart	0.1715	7.30E-03	0.1559	4.51E-03	0.1600	5.43E-03
Phoneme	0.2400	4.00E-04	0.1165	2.01E-04	0.1161	1.72E-04
Haberman	0.2696	2.00E-03	0.2458	3.36E-03	0.2491	2.40E-03
Titanic	0.2904	2.01E-02	0.2167	5.91E-04	0.2183	7.83E-04
Balance	0.1214	1.10E-03	0.0839	1.21E-03	0.0783	1.10E-03
Fertility	0.3130	7.47E-02	0.1850	1.05E-02	0.1250	2.50E-03
Skin&NonSkin	0.0761	2.21E-06	4.10E-04	1.53E-08	0.0004	1.60E-08
Wdbc	0.0678	1.10E-03	0.0387	5.98E-04	0.0397	6.97E-04
Australian	0.1980	1.80E-03	0.1222	1.30E-03	0.1233	1.20E-03
Twonorm	0.0216	2.83E-05	0.0219	2.78E-05	0.0219	2.72E-05
Magic	0.2733	5.06E-05	0.1921	8.34E-05	0.1923	7.93E-05
Ring	0.0209	2.20E-05	0.1131	1.16E-04	0.1131	9.98E-05
Tae	0.5595	1.39E-02	0.4365	1.36E-02	0.5132	1.67E-02
Contraceptive	0.5306	1.80E-03	0.4667	1.30E-03	0.4671	1.70E-03
Vehicle	0.5424	2.40E-03	0.2166	1.40E-03	0.2132	1.80E-03
Iris	0.0453	2.50E-03	0.0360	2.10E-03	0.0400	3.02E-03
Letter	0.3573	9.82E-05	0.0797	3.03E-05	0.0834	2.98E-05

Our results also show that GMM on Level1 data outperforms Decision Template, posting 10 wins and 0 loss. Superior results are reported on Bupa (30.22% vs. 33.48%), Haberman (24.58% vs. 27.79%), Fertility (18.5% vs. 45.2%), Skin&NonSkin (0.04% vs. 3.32%), Ring (11.31% vs. 18.94%) and Letter (7.97% vs. 11.33%).

GMM on Level1 data is also better than SCANN (5 wins and 1 loss). Likewise, GMMPCA is better than SCANN with 5 wins and 2 losses. We note that SCANN cannot be performed on 3 files, Skin, Balance and Fertility, due to the existence of equal column in the indicator matrix, which resulted in singular matrix [5]. Compared with MLR, both our approaches are equally competitive since both GMM on Level1 data and GMMPCA have 4 wins and 4 losses.

Table 5. Statistical tests compare GMM, GMMPCA with the Benchmarks

	Better	Competitive	Worse
GMM PCA Level1 vs. SCANN	5	11	2
GMM Level1 vs. SCANN	5	12	1
GMM PCA Level1 vs. Decision Template	11	9	1
GMM Level1 vs. Decision Template	10	11	0
GMM PCA Level1 vs. MLR	4	13	4
GMM Level1 vs. MLR	4	13	4
GMM PCA Level1 vs. best result from fixed rules	6	11	4
GMM Level1 vs. best result from fixed rules	5	13	3
GMM PCA Level1 vs. best result from base classifiers	7	11	3
GMM Level1 vs. best result from base classifiers	6	12	3
GMM PCA Level1 vs. GMM Level0	17	3	1
GMM Level1 vs. GMM Level0	16	4	1
GMM Level1 vs. GMM PCA Level1	2	17	2

Finally, we observe that the classification accuracy of GMM-PCA and GMM on Level1 data is comparable. However, since GMM-PCA can deal with situation when the number of observations is smaller than the dimension of data, it would be more applicable in many situations.

4 Conclusion and Future Work

We have introduced a novel approach which used GMM on Level1 data to combine results from base classifiers in a multi-classifier system. Experimental results on 21 UCI files have demonstrated the superiority of our method compared with several state-of-art combining algorithms. Future work will be to apply classifier and feature selection methods to further increase the classification accuracy of our approach.

References

1. Wolpert, D.H.: Stacked Generalization. *Neural Networks* **5**(2), 241–259 (1992)
2. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision Templates for Multi Classifier Fusion: An Experimental Comparison. *Pattern Recognition* **34**(2), 299–314 (2001)
3. Ting, K.M., Witten, I.H.: Issues in Stacked Generation. *Journal of Artificial In Intelligence Research* **10**, 271–289 (1999)
4. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3) (1998)
5. Merz, C.: Using Correspondence Analysis to Combine Classifiers. *Machine Learning* **36**, 33–58 (1999)
6. Zhang, L., Zhou, W.D.: Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition* **44**, 97–106 (2011)
7. Sen, M.U., Erdogan, H.: Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters* **34**, 265–274 (2013)
8. Moghaddam, B., Pentland, A.: Probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7) (July1997)

9. Liu, X.H., Liu, C.L.: Discriminative Model Selection for Gaussian Mixture Models for Classification. In: First Asian Conference on Pattern Recognition (ACPR), pp. 62–66 (2011)
10. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Press (2006)
11. Li, W., Prasad, S., Fowler, J.E.: Hyperspectral Image Classification Using Gaussian Mixture Models and Markov Random Fields. *IEEE Geoscience and Remote Sensing Letter* **11**(1) (January 2014)
12. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/datasets.html>
13. Rokach, L.: Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Journal of Computational Statistics & Data Analysis* **53**(12), 4046–4072 (2009)

Sentiment Classification of Chinese Reviews in Different Domain: A Comparative Study

Qingqing Zhou and Chengzhi Zhang^(✉)

Department of Information Management, Nanjing University of Science and Technology,
Nanjing 210094, China

breeze7zhou@gmail.com, zhangcz@njust.edu.cn

Abstract. With the rapid development of micro-blog, blog and other types of social media, users' reviews on the social media increase dramatically. Users' reviews mining plays an important role in the application of product information or public opinion monitoring. Sentiment classification of users' reviews is one of key issues in the review mining. Comparative study on sentiment classification results of reviews in different domains and the adaptability of sentiment classification methods is an interesting research topic. This paper classifies users' reviews in three different domains based on Support Vector Machine with six kinds of feature weighting methods. Experiment results in three domains indicate that different domains have their own characteristics and the selection of feature weighting methods should consider the domain characteristics.

Keywords: Sentiment classification · Chinese review · Support vector machine · Feature weighting

1 Introduction

Internet is becoming more central to our lives. The coexistence of openness, virtual and sharing of Internet makes it become a new platform for people expressing their emotions or views, such as product reviews, service reviews, blog reviews and etc [1].

As the number of various users' reviews on social media websites increases dramatically, how to organize the huge amount of data from reviews effectively has become a difficult problem for us. Sentiment analysis is to determine the users' attitudes for a particular topic or something, where attitudes can be their judgment, assessment or their (speech, writing) emotional state [2]. It is different from traditional text information processing as it focuses on the emotion expressed in the text. One of key issues of text sentiment analysis is text sentiment classification. Text sentiment classification is to judge the emotional tendencies or classify types of text [3].

In this paper, we mainly study sentiment classification in three different domains based on Support Vector Machine (SVM). Experiment results show that different domains have their own characteristics which affects the selection of feature weighting methods.

The rest of this paper is organized as follows. The second section describes the related work. In section 3, key technologies of SVM-based sentiment classification are described. Section 4 presents experiment results and section 5 concludes with a discussion of future work.

2 Related Works

Text sentiment classification is widely used, including: social public opinion analysis [4], product quality and service evaluation from consumers, harmful information filtering, books reviews, film and television reviews, blog reviews and so on.

In 2004, AAAI successfully held the conference with the theme "explore ideas and emotions in text" which accelerates the development of sentiment classification¹. Since then, research stage of sentiment classification is keeping on growing.

Sentiment classification consists of three steps: subjectivity classification, polarity classification and emotional intensity recognition [2]. Subjectivity Classification methods are based on supervised learning mostly. Wiebe used classifier based on Naïve Bayes with Boolean weighting in the task of subjectivity classification [5]. Complex syntactic structures can be used in the subjective classification [6][7]. Most of the current sentiment classification research focuses the polarity classification. Representative methods include: one is Turney's unsupervised method [8], another is Pang's supervised learning method [9]. Emotional intensity recognition can be implemented by the supervised learning methods. The main methods of this task can be divide into the following three categories: (1) the multi-classification method: Lin divided emotional intensity in the sentence into five levels and used LSPM to distinguish intensity [10], (2) the regression method: Pang & Li used SVM regression method to identity emotional intensity [11] and (3) the sequence annotation method: Liu put forward the sentence sentiment degree analysis model based on cascaded CRFs model [12].

Currently, performance of sentiment classification results in different domains is various. However, there is no systematic and comprehensive study about this topic. This paper studies how to use machine learning to do sentiment classification automatically and compare the performance of different domains with different feature weighting method.

3 Framework and Key Technologies of SVM-Based Sentiment Classification

3.1 Feature Selection

Reviews in Chinese must be segmented before the feature selection. This paper applies improved maximum matching word segmentation algorithm (MMSEG) [13] to segment reviews in Chinese. MMSEG is a dictionary-based word segmentation algorithm.

Not all words are useful for sentiment classification, so feature selection is necessary. The representative words can be extracted from text according to feature weighting methods [14]. There are many classical feature selection methods, such as TF, DF, IG, MI, CHI etc [15]. Hwee's experiment results show that the CHI is the best feature selection method according to the performance of classification [16]. Chen compared existing feature extraction methods and proved that no feature selection method is applicable to all or most of the experimental corpora [17]. Above all, we choose the CHI method for feature selection in this paper.

¹ <http://www.aaai.org/Press/Proceedings/aaai04.php>

3.2 Feature Weighting Methods

The performance of feature weighting methods affects the classification accuracy directly. Classical feature weighting methods includes Boolean weights, TF, IDF, TF*IDF and so on.

Deng replaced IDF with CHI, and experiment result shows that TF-CHI performed better than TF-IDF in the task of SVM-based text classification [18]. TF-CHI can be computed via formula 1:

$$w_i = tf(t_i) * CHI(t_i, C_j) \quad (1)$$

Lan proposed a weighting method, namely TF*RF (where RF means Relevance Frequency) and experiment result proves that it has better performance than TF*IG and other methods [19]. RF can be calculated through formula 2:

$$rf = \log \left(2 + \frac{a}{c} \right) \quad (2)$$

Where, a denote co-occurrence number of the feature and class, c means the number of feature appears but class does not appear.

The feature weighting methods include Boolean weight, TF, $\log(TF)$, TF * IDF, TF * CHI and TF * RF is used in this paper. These methods are compared according to the performance of sentiment classification.

3.3 Parameters Selection and Optimization of SVM Model

LIBSVM² is used to classify the reviews in this paper. Two most important parameters in LIBSVM are C and γ . The C is the sample misclassification penalty factor. The larger the C is, the smaller the error tolerates [14]. Whether the C is too large or too small will affect the generalization ability of the model. The γ is the parameter come from the RBF kernel function. The larger the γ is, the more support vectors. The number of support vectors affects the speed of the model training and prediction directly.

3.4 Classification Results Determination

Classification results in this paper consist of two parts: predicted category and membership degree. The larger the membership degree is, the greater confidence that the sample belongs to the class [14]. Membership degree's is computed by the following formula:

$$M = \frac{\sum S_i}{2 * K} + \frac{K_s}{2 * K} \quad (3)$$

Where, S_i denotes the score of support discrimination classes, K_s means the number of support discrimination classes, K means the number of all categories. Membership degree is used to improve accuracy rate as the credibility of using category labels alone as the classification results is low. The membership degree algorithm in the paper is the one-against-one algorithm [20].

² <http://baike.baidu.com/view/598089.htm>

4 Experiments and Result Analysis

4.1 Experimental Data

We use blog reviews of ScienceNet³, hotel reviews of Ctrip⁴ and book reviews of Dangdang⁵ as experiment data, each type of data contains training set and test set as shown in table 1. Table 2 shows experiment samples.

Table 1. Experiment data set

Domain	Training set	Positive	Negative	Test set
Blog Reviews	950	600	350	2,800
Hotel Reviews	1,000	500	500	3,633
Book Reviews	1,000	500	500	2,870

Table 2. Experiment data sample

Category		Samples
Blog Reviews	Positive reviews	写得好！顶一个！
	negative reviews	你是帮倒忙的吧，以后写文章不要这样。。。。
Hotel Reviews	Positive reviews	环境很好，地点很方便，服务也很好，下回还会住的！
	negative reviews	缺点：太多了;1,设施太陈旧了，地毯到处都是黑污渍，房间有股怪味，虽然3星级不能要求太多，但也不能比经济型酒店还差吧...
Book Reviews	Positive reviews	这本书不错，读过以后，才知道该如何面对与解决自己或周边关系存在的问题。这是学校里没有的。当然，这是要靠自己去领悟书中思想...
	negative reviews	这是一本捡别人漏沟水的书，非常后悔买了这么一本毫无看头的书，请大家别再上当了！！！！

³ <http://blog.sciencenet.cn/>

⁴ <http://hotels.ctrip.com/>

⁵ <http://www.dangdang.com/>

4.2 SVM Model Training

We use MMSEG algorithm to segment the reviews in Chinese, CHI method for feature selection, six different kinds of feature weighting methods, namely: Boolean weight, TF, log (TF) TF-IDF, TF- RF, TF-CHI, LIBSVM for model training. We train models in turn and get 18 different models.

4.3 Experiment Results Evaluation Method

The evaluation indicators include: Precision, Recall and F1 value [21]. The contingency table for results evaluation of classification is shown in table 3. contingency table for results evaluation of classification

Table 3. Contingency table for results evaluation of classification

		Prediction Classification	
		P(Positive)	N(Negative)
Real classification	P	TP(Ture ositive)	FN(False Negative)
	N	FP(False Positive)	TN(Ture Negative)

These evaluation indicators are calculated via the following formula respectively:

$$(a) \text{ Recall} \quad R = \frac{TP}{TP+FP} \quad (4)$$

$$(b) \text{ Precision} \quad P = \frac{TP}{TP+FN} \quad (5)$$

$$(c) F_1 \text{ value} \quad F_1 = \frac{2*P*R}{P+R} \quad (6)$$

For all classes:

$$(a) \text{ Macro Recall} \quad \text{MacroR} = \frac{1}{n} \sum_{j=1}^n R_j \quad (7)$$

$$(b) \text{ Macro Precision} \quad \text{MacroP} = \frac{1}{n} \sum_{j=1}^n P_j \quad (8)$$

(c) MacroF₁ value

$$\text{MacroF}_1 = \frac{2*\text{MacroP}*\text{MacroR}}{\text{MacroP}+\text{MacroR}} \quad (9)$$

4.4 Experiment Results Analysis

In order to analyze the experiment results, we annotated the polarity of testing corpus manually. The testing set of blog reviews contains 2,100 positive reviews and 700 negative reviews. The test set of hotel reviews contains 1,702 positive reviews and 1,921 negative reviews. The test set of book reviews contains 1,420 positive reviews and 1,450 negative reviews.

4.5 Results Analysis of the Same Corpus with Different Feature Weighting

(a) Blog reviews of ScienceNet

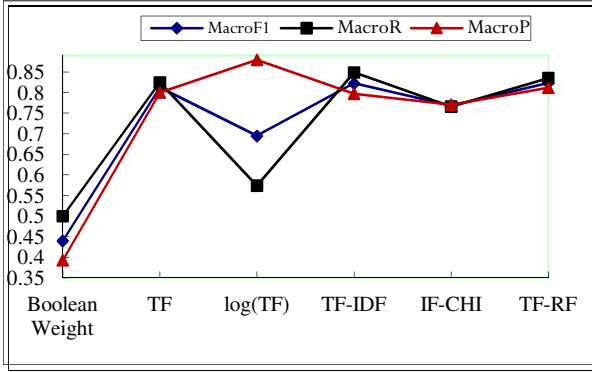


Fig. 1. Blog reviews of ScienceNet

From figure 1 we can find that for blog reviews of ScienceNet, TF-RF has the best classification performance, followed by TF-IDF, Boolean weighting method does worst.

(b) Hotel reviews of Ctrip

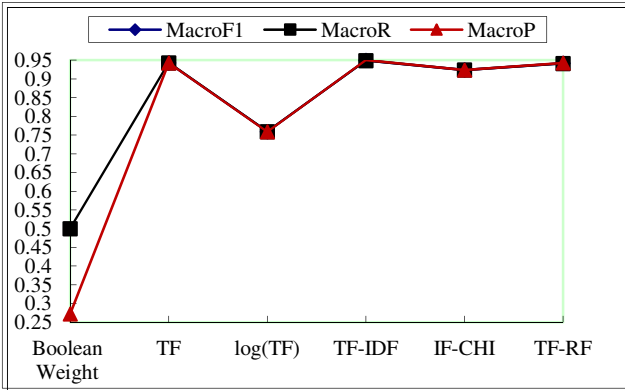


Fig. 2. Hotel reviews of ctrip

Figure 2 shows that for hotel reviews of Ctrip, the best classification is done by TF-IDF, followed by TF, Boolean weighting method performs worst.

(c) Book reviews of Dangdang

From Figure 3 we can find that that for book reviews of Dangdang, TF-IDF has best classification performance, followed by TF, Boolean weighting method classifies worst.

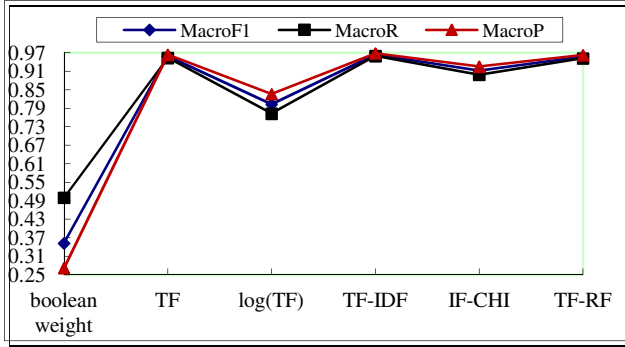


Fig. 3. Book reviews of Dangdang

4.6 Results Analysis of Different Corpus with the Same Feature Weighting

Figure 4 shows that IF-IDF has optimal classification performance in the three different reviews corpus, followed by TF-RF and TF, TF-CHI and Log (TF) perform in general, the classification performance of Boolean weighting method is the worst.

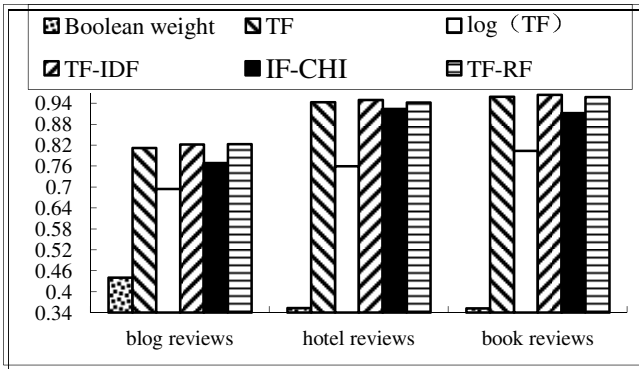


Fig. 4. Three reviews corpus evaluation

We speculated the results have the following reasons:

(a) The average length of reviews

We calculated the average length of three kinds of reviews, and got that the average length of blog reviews is 14.61 words, hotel reviews’ average length is 85.87 words and book reviews’ average length is 104.03 words. We speculate that TF, TF-IDF, TF-RF, Log(TF) and Boolean weight are linearly related to the average length of reviews, the first four is positive correlation, the last one is negative correlation. TF-CHI is curve related to the average length of reviews, and it forms a convex function.

(b) The ratio of positive and negative reviews

We calculated the ratio of positive and negative reviews of the three domains, we got that the ratio of blog reviews is 1:0.33, hotel reviews’ ratio is 1:1.13, book reviews’ ratio is 1:1.02. We speculate that TF, TF-IDF, TF-RF, Log(TF) and Boolean weight are curve

related to the ratio of reviews, the first four form convex functions, the last one forms a Concave function. TF-CHI is linearly related to the ratio of reviews and it is positive correlation.

(c) Other reasons

In addition to reasons above, sentiment classification is associated with some emotional factors such as its own characteristics of each domain, the review language characteristics and viewpoint holders. It is difficult to do quantitative comparison of these reasons' effect.

4.7 Results Analysis of the Same Corpus with Different Threshold

As IF-IDF, TF-RF and TF perform better than several other feature weighting methods, we analyze the performance of the three methods in different thresholds only.

(a) Blog reviews of ScienceNet

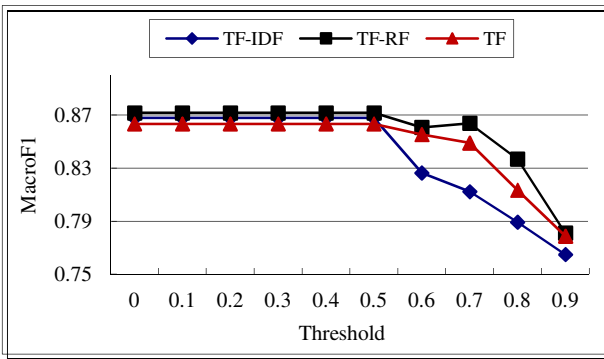


Fig. 5. Blog reviews of ScienceNet

From figure 5 we can find that TF-IDF, TF-RF and TF perform best when threshold less than or equal to 0.5, where TF-RF has optimal performance, followed by TF-IDF.

(b) Hotel reviews of Ctrip

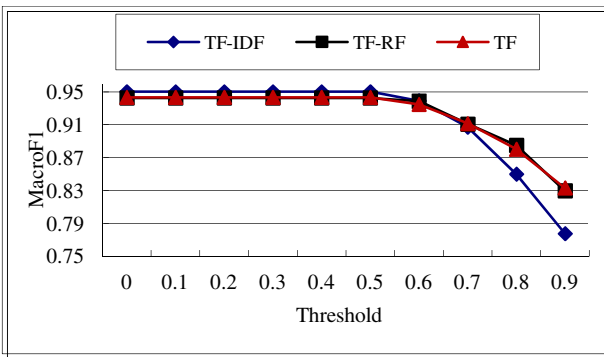


Fig. 6. Hotel reviews of Ctrip

Figure 6 shows that TF-IDF, TF-RF and TF perform best when threshold less than or equal to 0.5, in which TF-IDF has best performance, followed by TF.

(c) Book reviews of Dangdang

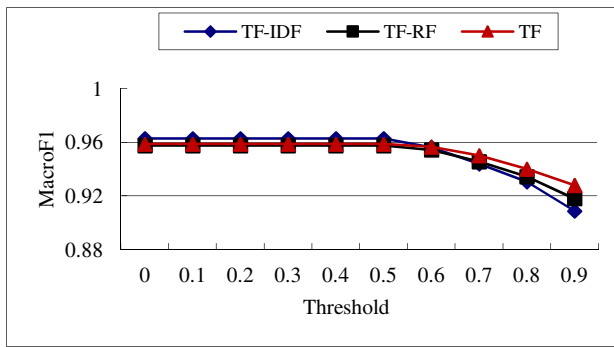


Fig. 7. Book reviews of Dangdang

From figure 7 we can see that TF-IDF, TF-RF and TF classify best when threshold less than or equal to 0.5, where TF-IDF has optimal performance, followed by TF.

Based on the above threshold performance, we can draw the conclusion that TF-IDF, TF-RF and TF perform best when threshold less than or equal to 0.5.

5 Conclusion and Future Works

5.1 Conclusion

Sentiment classification based on SVM is performed and classification results are compared in different domains. By analyzing the experiment results, adopting feature weighting method TF-RF attained the optimal performance for blog reviews of ScienceNet while TF-IDF works best for hotel reviews of Ctrip and book reviews of Dangdang. Overall, we found that feature weighting method TF-IDF performed the best, followed by TF, TF-RF. For different thresholds, we concluded that TF-IDF, TF-RF and TF classified best when thresholds are less than or equal to 0.5.

We concluded that different domains have their own characteristics, the domain characteristics are needed to be taken into account for the selection of feature weighting methods.

5.2 Future Works

In the future, our aim is to improve the performance of sentiment classification by the following approaches.

- (a) We will expand the amount of experiment data.
- (b) We will increase the equality of experiment data in three domains.
- (c) We will add more feature selection algorithms, such as IG and MI.
- (d) We will replace polarity classification with multi-classification for Emotion intensity recognition.

Acknowledgements. This work was supported in part by the Fundamental Research Funds for the Central Universities (No. 30920130132013), National Natural Science Foundation of China (No.70903032) and Project of the Education Ministry's Humanities and Social Science (No. 13YJA870020).

References

1. Wang, S.: Text sentiment classification research on web-based reviews. Shanghai University, Shanghai, pp. 1–5 (2008) (in Chinese)
2. Wang, H., Liu, X., Yin, P., Liao, Y.: Web text sentiment classification research. *Scientific and Technical Information* **29**(5), 931–938 (2010) (in Chinese)
3. Zhang, Y.: Text sentiment classification research. Beijing Jiaotong University, Beijing, pp. 1–10 (2010) (in Chinese)
4. Agrawal, R., Rajagopalan, S., Srikant, R., Xu, Y.: Mining newsgroups using networks arising from social behavior. In: *Proceeding of the 12th WWW Conference*, Budapest, Hungary, pp. 529–535 (2003)
5. Wiebe, J.M., Bruce, R.F., O'Hara, T.P.: Development and use of a gold-standard data set for subjectivity classifications. In: *Proceedings of the Association for Computational Linguistics*, pp. 246–253. College Park, Maryland (1999)
6. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp. 105–112. ACL, USA (2003)
7. Wiebe, J., Wilson, T., Bruce, R.F., Bell, M., Martin, M.: Learning subjective language. *Computational Linguistics* **30**(3), 277–308 (2004)
8. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, pp. 417–424 (2002)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp. 79–86. ACL, USA (2002)
10. Lin, W.H., Wilson, T., Wiebe, J., et al.: Which side are you on? Identifying perspectives at the document and sentence levels. In: *Proceedings of the 10th Conference on Computational Natural Language Learning*, NY, USA, pp. 109–116 (2006)
11. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of Conference on Association for Computational Linguistics*, Michigan, pp. 115–124 (2005)
12. Ni, X., Xue, G.R., Ling, X., et al.: Exploring in the weblog space by detecting informative and affective articles. In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 281–290. ACM (2007)
13. Tsai, C.H.: MMSEG: a word identification system for mandarin Chinese text based on two variants of the maximum matching algorithm (2000). <http://www.geocities.com/ha0150/mmseg/>
14. Zhang, Z.: Tmsvm Reference Documents. [tmsvm.googlecode.com/svn/trunk/Tmsvm Reference Documents \(v1.1.0\).docx](http://tmsvm.googlecode.com/svn/trunk/Tmsvm%20Reference%20Documents%20(v1.1.0).docx) Accessed: (May 1, 2013) (in Chinese)
15. Yang, Y., Pederson, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: *Proceedings of the Fourteenth International Conference*, pp. 412–420 (1997)
16. Ng, H.T., Goh, W.B., Leong, K.: Feature selection, perceptron learning, and a usability case study for text categorization. *ACM SIGIR Forum* **31**(SI), 67–73 (1997)

17. Chen, T., Xie, Y.: Feature dimension reduction methods for text classification. *Scientific and Technical Information* **24**(6), 690–695 (2005)
18. Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Zhang, M., Li, L.-Y., Xie, K.-Q.: A comparative study on feature weight in text categorization. In: *Proceedings of the 6th Asia-Pacific Web Conference*, Hangzhou, China, pp. 588–597 (2004)
19. Lan, M., Tan, C.-L., Low, H.-B.: Proposing a New TermWeighting Scheme for Text Categorization. In: *Proceedings of AAAI Conference on Artificial Intelligence*, Boston, Massachusetts, pp. 763–768 (2006)
20. Liu, B., Hao, Z., Xiao, Y.: Interactive iteration on one against one classification algorithm. *Pattern Recognition and Artificial Intelligence*, **21**(4):425–431 (2008) (in Chinese)
21. Salton, G., McGill, M.J.: *Introduction to Modern Information Retrieval*. McGraw Hill Book Co. (1983)

A Computer-Aided System for Classification of Breast Tumors in Ultrasound Images via Biclustering Learning

Qiangzhi Zhang¹, Huali Chang¹, Longzhong Liu², Anhua Li², and Qinghua Huang¹(✉)

¹ School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

qhhuang@scut.edu.cn

² The Cancer Center of Sun Yat-sen University, Guangzhou, China

Abstract. The occurrence of Breast cancer increases significantly in the modern world. Therefore, the importance of computer-aided recognition of breast tumors also increases in clinical diagnosis. This paper proposes a novel computer-aided diagnosis (CAD) method for the classification of breast lesions as benign or malignant tumors using the biclustering learning technique. The medical data is graded based on the sonographic breast imaging reporting with data system (BI-RADS) lexicon. In the biclustering learning, the training data is used to find significant grading patterns. The grading pattern being learned is then applied to the test data. The k-Nearest Neighbors (k-NN) classifier is used as the classifier of breast tumors. Experimental results demonstrate that the proposed method classifies breast tumors into benign and malignant effectively. This indicates that it could yield good performances in real applications.

Keywords: Computer-aided diagnosis (CAD) · Breast cancer · Biclustering · K-NN · BI-RADS

1 Introduction

Breast cancer has a high death rate and kills millions of women all over the world each year. However, there is still lack of effective treatment yet. So, early detection becomes very important to increase the survival rate of patients [1]. Since ultrasound imaging (USI) has advantages of being non-radiative, non-invasive, real-time and low-cost, it becomes one of the most important methods for breast cancer diagnosis [2]. To reduce the subjective dependency and improve the diagnostic accuracy, computer-aided diagnosis (CAD) system is badly needed to obtain reliable diagnostic results.

Generally, the traditional CAD system for breast cancer based on the USI involves four stages, i.e. image preprocessing, segmentation, feature extraction and selection, and classification. Owing to the complex imaging environment and principle, the USI unavoidably contains a lot of artifacts and noises which leads to speckles, shadows and low contrast. Therefore, a preprocessing procedure for speckle reduction is necessary. Image segmentation is one of key procedures in the CAD system. But it is worth to mention that it is often difficult to segment breast ultrasound (BUS) images due to the speckles and low contrast which are inherent in BUS images. It is also critical to find a feature set of breast tumor which can distinguish between benign and malignant

accurately. However, the traditional CAD system usually performs the diagnosis based on texture, color, Doppler and morphologic features of breast tumors that have some limitations on breast tumor diagnosis. Finally, it is very time consuming to train a good classifier with high classification accuracy. In such a CAD framework, the final diagnostic results will be unreliable if any one of the four stages does not perform well and the whole process is complicated.

A number of CAD systems for breast cancer based on the USI have been presented in recent years. Ding et. al. [3] applies multiple-instance learning method to classify the tumors in BUS images. Chang et. al. [4] proposes an automatic tumor segmentation and a shape analysis technique to improve the distinction between benign and malignant breast tumors. Garra et. al. [5] uses quantitative analysis of ultrasound image texture to improve the ability of ultrasound to distinguish benign from malignant breast tumors. In recent years, Doppler spectral analysis serves as a useful tool in distinguishing between benign and malignant breast tumors. Some studies have been conducted based on the Doppler spectral analysis to achieve improved performance for the classification of breast tumors [6], [7], [8]. A combination of wavelet transform and neural networks techniques is proposed to improve the accuracy of breast tumor classification [9].

The breast imaging reporting and data system (BI-RADS) lexicons are designed to aid radiologists in describing abnormalities for sonographic breast findings. In recent years, many studies have shown that breast tumor analysis can achieve better performance based on the BI-RADS lexicons [10], [11]. On the other hand, biclustering has emerged as one of the most popular tools for data mining. It identifies object with the same attribute or the same function which would be more beneficial for mining important information.

In this paper, the proposed method adopts a new CAD framework for the classification of breast cancers as benign or malignant. Methods for grading the input image data under features recommended by the sonographic BI-RADS lexicons [12] and biclustering grades using the prior label information for mining diagnostic patterns are proposed. Finally, the k-NN classifier is applied to perform classification of tumor types.

This paper is organized as follows. Section 2 provides a brief introduction of the novel CAD system being proposed. Section 3 presents and discusses the experimental results of the proposed method. We conclude this paper in Section 4.

2 Proposed Methods

In our study, each tumor in BUS image is described by 17 features (from the sonographic BI-RADS lexicon) which are graded with 4 grades by experienced clinicians. Then, the biclustering learning algorithm is applied to mine useful information in feature matrix which we get by grading. At last, the k-NN classifier is employed to perform classification.

2.1 Grading of Medical USI Data

The American College of Radiology (ACR) commission has published an ultrasound lexicon for breast cancer called the breast imaging reporting and data system

(BI-RADS) in 2003 to standardize the reporting of sonographic breast cancer finding and as a communication tool for clinicians [13]. This lexicon includes descriptors of features such as mass shape, orientation, lesion boundary, echo pattern margin, posterior acoustic features, and blood supply, etc [12]. Several studies have confirmed the utility of these features described in the sonographic BI-RADS lexicon in distinguishing benign and malignant breast tumors [14], [15], [16].

Many studies have shown that breast tumor in BUS image has a number of sonographic characteristics significantly different for malignant and benign tumors, which allows the classification as either malignant or benign [12], [17], [18]. For example, malignant tumor is commonly hypoechoic lesion with ill-defined border which is “taller than broader”, with spiculate and angular margin, duct extension and branch pattern. However, benign tumors usually tend to have a smooth and well circumscribed border, three or fewer gentle lobulations and thin echogenic capsule. Additionally, a BUS image which shows a malignant nodule commonly has internal calcification, posterior shadowing and an abundant blood supply in or around the tumor. However, these characteristics rarely appear or have a comparatively slight degree in BUS image with benign nodules.

Based on above-mentioned, we choose 17 features (according to the sonographic BI-RADS lexicon) to describe each tumor and grade them with 4 grades. Taking the feature of blurred boundary as an example, clinician would give it a grade 1, 2, 3 or 4 (corresponding to not blurred, slightly blurred, relatively blurred and extremely blurred, respectively) based on a given BUS image. Thus, a two-dimensional feature matrix is constructed, and each row represents the features of a BUS image with label (benign or malignant), each column represents the feature grades given by experienced clinicians. The features being used according to the sonographic BI-RADS lexicon are listed in Table 1.

2.2 Biclustering Learning Algorithm

Biclustering algorithm is a data mining method to find statistically significant sub-matrix (also called bicluster) in a data matrix [20], [21], [22], [23]. After constructing a feature matrix by grading, biclustering learning algorithm is performed to get biclusters which will be the training set of k-NN classifier. Details are as follows.

We first apply a traditional agglomerative hierarchical clustering (HC) method [19] on each column of the two-dimension feature matrix A (for convenience, A represents the feature matrix later on) to get bicluster seeds. Given that A has R rows and C columns, bicluster seeds of every column are formulated by:

$$[C_s(i, j), N_{c_l}(j)] = HC_ (j, \tau), j = 1, \dots, C \quad (1)$$

where HC is the agglomerative hierarchical clustering algorithm, τ is the distance threshold for HC , $C_s(i, j)$ and $N_{c_l}(j)$ are the i th bicluster seed and the number of clusters in the j th column, respectively.

Table 1. The features being used by proposed system according to the sonographic bi-rads

Features		
1. Shape	2. Orientation	3. Margin
4. Indistinct	5. Angular	6. Microlobulated
7. Spiculated	8. The boundary	9. Echogenicity
10. Internal echo pattern	11. Posterior echo pattern	12. Blood supply
13. Effect on surrounding tissue ---Edema	14. Effect on surrounding tissue --- Architectural distortion	15. Effect on surrounding tissue ---Ducts
16. Calcification in mass	17. Calcifications out of mass	

As aforementioned, the detected clusters from all columns are treated as bicluster seeds. After all the bicluster seeds are discovered, the next step is to expand each of them along the column direction according to the mean-square-residue (MSR) score [20] of the sub-matrix. MSR score has been mostly used as a natural assessment criterion for the quality of bicluster. Whether or not to delete a column in the refining step is depended on if the newly formed sub-matrix satisfies the criterion of MSR score (i.e. the MSR score is less than a preset threshold δ) [21].

Given an $R \times C$ matrix, MSR score is formulated as follows:

$$H(I, J) = \sum_{i \in I, j \in J} \frac{(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2}{|I||J|} \quad (2)$$

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, \quad a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij} \quad (3)$$

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} \quad (4)$$

where a_{ij} represents the element value in the matrix corresponding to row i and column j , $H(I, J)$ is the value of MSR score of bicluster. Only if the MSR score of newly formed sub-matrix is less than the preset threshold δ , then accept it as a valid bicluster.

Finally, this study defines a support-based metric as a measure to help make the classification after all valid biclusters have been detected. We make use of the labels of ultrasound images to calculate the support of biclusters discovered in the previous step, which is expressed as:

$$Spt(s_m) = \frac{R_m}{R_i}, Spt(s_b) = \frac{R_b}{R_i} \quad (5)$$

where $Spt(s_m)$ and $Spt(s_b)$ denote the supports of malignant and benign, respectively, R_m denotes the number of rows with malignant label in the i th bicluster, R_b denotes the number of rows with benign label, and R_i is the total number of rows of the i th bicluster.

To make a classification decision, we have to know the maximum of the supports, as follows:

$$S(\bullet) = \max(Spt(s_m), Spt(s_b)) \quad (6)$$

where S denotes the maximum value among $Spt(s_m)$ and $Spt(s_b)$, indicating that the type of the bicluster is determined by the signal with the largest support.

In order to obtain more reliable biclusters, we select the bicluster whose S is larger than a pre-defined threshold S_{pre} as a useful classification signal. S_{pre} is set to 0.8 in this paper. The detected biclusters with support information are grouped into three types of signals: benign, malignant and no-action signals. For instance, given that $Spt(s_m) = 0.85$, $Spt(s_b) = 0.15$, respectively, the bicluster is regarded as a malignant signal and given a malignant label, because $S = \max(Spt(s_m), Spt(s_b)) = Spt(s_m) = 0.85$ is greater than S_{pre} . For another instance, if the supports of malignant and benign are 0.4, 0.6, respectively, the bicluster is classified as a no-action signal, due to $S = 0.6$ which is below the pre-defined threshold 0.8. Additionally, we remove the useless bicluster whose S is less than S_{pre} . Thus we can achieve several meaningful signals which are more beneficial to classification.

2.3 K-NN Classifier

In pattern recognition, k-NN classifier is one of the most popular classifiers [24]. The idea of k-NN algorithm is quite simple and straightforward. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors measured by a distance function.

After previous steps, we obtain several useful biclusters from the training set with label information. A bicluster is translated into a classification rule by averaging each column. The output classification rule is a vector where the value of an element is the mean of the corresponding indicators values over the BUS images included by the bicluster. Then, calculate the modified Euclidean distances between object in test set and all valid biclusters acquired from training set. If $X=(x_1, x_2, \dots, x_m)$ and $Y=(y_1, y_2, \dots, y_n)$ ($m \leq n$, $n = 17$) are two samples taken from the training set and testing set, separately. The modified Euclidean distance function used to compute the similarity between two samples is defined as following:

$$D(X, Y) = \frac{1}{m} \sqrt{\sum_{i \in m} (x_i - y_i)^2} \quad (7)$$

where x_i, y_i are grades of features given by clinician.

We set k to be 3 in our study (also try other values, but get best performance when k is set to 3). In order to overcome the problem of small dataset, we adopt the leave-one-out cross validation method to evaluate the performance of our method. The advantage of leave-one-out cross validation is that each instance in one dataset can be used for both training and testing.

The flow diagram of the proposed system is shown in Figure 1.

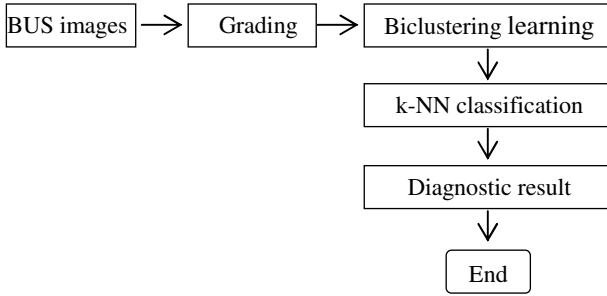


Fig. 1. The flow diagram of the proposed system

3 Experimental Results

The proposed system is developed by VC++6.0 (Microsoft Co. Ltd., USA) and evaluated using a 51 BUS images dataset (29 malignant and 22 benign). The BUS images are provided by the Cancer Center of Sun Yat-sen University and imaged by an IU22 SonoCT System (Philips Medical Systems) with a L12-5 50mm Broadband Linear Array at the imaging frequency of 7.1MHz.

To assess the performance of the proposed method, the specificity, sensitivity and accuracy are used to evaluate the performance of the classification capability of the proposed method. Define the number of correctly classified benign and malignant tumors as true negative (TN) and true positive (TP), respectively. The number of incorrectly classified benign and malignant tumors as false negative (FN) and false positive (FP), respectively.

$$Specificity(TNF) = \frac{TN}{TN + FP} \quad (8)$$

$$Sensitivity(TPF) = \frac{TP}{TP + FN} \quad (9)$$

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (10)$$

In order to validate the accuracy of the proposed scheme, it is compared with the other two systems proposed in [4], [8]. Table 2 illustrates the quantitative analysis of these three methods.

Table 2. The descriptive statistics (syst 1 [ref. 4], syst 2 [ref. 8])

<i>Method</i>	<i>Accuracy</i>	<i>Specificity</i>	<i>Sensitivity</i>
Syst 1	90.95 %	92.50 %	88.89 %
Syst 2	85.60%	79.50%	97.60%
Our syst	94.12%	95.45%	93.10%

From Table 2, we can see that our method achieves the highest overall classification accuracy, specificity, and sensitivity (94.12%, 95.45% and 93.10%). It is obvious that the proposed method has better performance than that of [4] and [8]. Generally speaking, compared with the traditional CAD systems, our method further improves the accuracy of breast tumor classification.

4 Discussions and Conclusions

In this paper, an effective classification system for breast tumor is proposed. Firstly, tumors are characterized according to the BI-RADS lexicon for breast ultrasound and each feature is graded from 1 to 4. Then, a biclustering learning algorithm is applied to find meaningful local-coherent patterns. Detected patterns with support information, namely supervised biclusters, are used as classification basis and grouped into benign, malignant and invalid sets. To the best of our knowledge, this is the first attempt to take the advantage of both biclustering algorithm and supervised learning for CAD system of breast cancer diagnosis. Finally, the k-NN classifier is adopted to classify breast tumor as benign or malignant based on corresponding feature vector and supervised biclusters. Syst. 1 extracts the solidity morphologic features from B-Mode ultrasound images and then a SVM classifier is employed to classify the tumor as benign or malignant. In Syst. 2, textural features, morphologic features and color Doppler features are extracted from the B-Mode and the color Doppler images. Those features are then used to classify benign and malignant tumors. Compared with systems proposed in [4] and [8], a novel CAD method using a new biclustering learning technique for classification of breast tumor as benign or malignant is proposed.

Quantitative experimental results demonstrate that the proposed system significantly improves the accuracy of breast tumor classification. However, the dataset being used in our experiment only has a quite limited number of samples. Accordingly, we will test our method on an expanded dataset with more samples and different tumor types to obtain more reliable results in our future work. With further efforts to improve the proposed scheme, it is expected that this scheme will become more valuable to radiologists and be used as a useful diagnostic aid in real clinical applications.

Acknowledgements. This work is supported by National Natural Science Funds of China (No. 61372007), Natural Science Funds of Guangdong Province (No. S2012010009885), the Fundamental Research Funds for the Central Universities (No. 2014ZG0038), and Projects of innovative science and technology, Department of Education, Guangdong Province (No. 2013KJCX0012).

References

1. Landis, S.H., Murray, T., Bolden, S., Wingo, P.A.: Cancer statistics. *CA: A Cancer Journal for Clinicians* **48**(1), 6–29 (1998)
2. Chen, D., Chang, R., Wu, W., Moon, W.K., Wu, W.: 3-D breast ultrasound segmentation using active contour model. *Ultrasound in Medicine and Biology* **29**(7), 1017–1026 (2003)
3. Ding, J., Cheng, H.D., Huang, J., Liu, J., Zhang, Y.: Breast Ultrasound Image Classification Based on Multiple-Instance Learning. *Journal of Digital Imaging* **25**(5), 620–627 (2012)
4. Chang, R., Wu, W., Moon, W.K., Chen, D.: Automatic ultrasound segmentation and morphology based diagnosis of solid breast tumors. *Breast Cancer Research and Treatment* **89**(2), 179–185 (2005)
5. Garra1, B.S., Krasner, B.H., Horii, S.C., Ascher, S., Mun, S.K., Zeman, R.K.: Improving the Distinction between Benign and Malignant Breast Lesions: The Value of Sonographic Texture Analysis. *Ultrasonic Imaging* **15**(4), 267–285 (2002)
6. Kuo, W., Chen, D.: Classification of benign and malignant breast tumors using neural networks and three-dimensional power Doppler ultrasound. *Ultrasound in Obstetrics & Gynecology* **32**(1), 97–102 (2008)
7. Diao, X., Wang, T., Yang, Y., Chen, S.: Computer-aided diagnosis of breast tumor based on B-mode ultrasound and color Doppler flow imaging. In: *Proceeding of BMEI 2009 Conference, Tianjin*, pp. 1–5 (October 2009)
8. Liu, Y., Cheng, H.D., Huang, J., Zhang, Y., Tang, X., Tian, J., Wang, Y.: Computer Aided Diagnosis System for Breast Cancer Based on Color Doppler Flow Imaging. *Journal of Medical Systems* **36**(6), 3975–3982 (2012)
9. Chen, D., Chang, R., Kuo, W., Chen, M.: Diagnosis of breast tumors with sonographic texture analysis using wavelet transform and neural networks. *Ultrasound in Medicine & Biology* **28**(10), 1301–1310 (2002)
10. Jales, R.M., Sarian, L.O., Torresan, R., Marussi, E.F., Alvares, B.R., Derchain, S.: Simple rules for ultrasonographic subcategorization of BI-RADS (R)-US 4 breast masses. *European Journal of Radiology* **82**(8), 1231–1235 (2013)
11. Park, C.S., Lee, J.H., Yim, H.W., Kang, B.J., Kim, H.S., Jung, J.I., Jung, N.Y., Kim, S.H.: Observer agreement using the ACR breast Imaging reporting and data system (BI-RADS)-Ultrasound, first edition (2003). *Korean Journal of Radiology* **8**(5), 397–402 (2007)
12. Mendelson, E.B., Berg, W.A., Merritt, C.R.B.: Toward a standardized breast ultrasound lexicon, BI-RADS: ultrasound. *Seminars in Roentgenology* **36**(3), 217–225 (2001)
13. Levy, L., Suissa, M., Chiche, J.F., Teman, G., Martin, B.: BIRADS ultrasonography. *European Journal of Radiology* **61**(2), 202–211 (2007)
14. Hong, A.S., Rosen, E.L., Soo, M.S., Baker, J.A.: BI-RADS for sonography: positive and negative predictive values of sonographic features. *American Journal of Roentgenology* **184**(4), 1260–1265 (2005)
15. Thomas, A., Thickman, D., Rapp, C.L., Dennis, M.A., Parker, S.H., Sisney, G.A.: Solid breast nodules: use of sonography to distinguish between benign and malignant lesions. *Radiology* **196**(123) (1995)

16. Rahbar, G., Sie, A.C., Hansen, G.C., Prince, J.S., Melany, M.L., Reynolds, H.E., Jackson, V.P., Sayre, J.W., Bassett, L.W.: Benign versus malignant solid breast masses: US differentiation. *Radiology* **213**(3), 889–894 (1999)
17. Heinig, J., Witteler, R., Schmitz, R., Kiesel, L., Steinhard, J.: Accuracy of classification of breast ultrasound findings based on criteria used for BI-RADS. *Ultrasound in Obstetrics & Gynecology* **32**(4), 573–578 (2008)
18. Mainiero, M.B., Goldkamp, A., Lazarus, E., Livingston, L., Koelliker, S.L., Schepps, B., Mayo-Smith, W.W.: Characterization of Breast Masses With Sonography Can Biopsy of Some Solid Masses Be Deferred? *Journal of Ultrasound in Medicine* **24**(2), 161–167 (2005)
19. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1), 4–37 (2000)
20. Cheng, Y.Z., Church, G.M.: Biclustering of expression data. In: *Proceedings of ISMB 2000 Conference*, pp. 93–103 (August 2000)
21. Huang, Q.H., Tao, D.C., Li, X.L., Jin, L.W., Wei, G.: Exploiting Local Coherent Patterns For Unsupervised Feature Ranking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **41**(6), 1471–1482 (2011)
22. Huang, Q.H.: Discovery of time-inconsecutive co-movement patterns of foreign currencies using an evolutionary biclustering method. *Applied Mathematics and Computation* **218**(8), 4353–4363 (2011)
23. Huang, Q.H., Tao, D.C., Li, X.L., Liew, A.W.C.: Parallelized evolutionary learning for detection of biclusters in gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**(2), 560–570 (2012)
24. Wu, X.D., Kumar, V., Quinlan, J.R., et al.: Top 10 algorithms in data mining. *Knowledge and Information Systems* **14**(1), 1–37 (2008)

An Improved Approach to Ordinal Classification

Donghui Wang, Junhai Zhai^(✉), Hong Zhu, and Xizhao Wang

Key Laboratory of Machine Learning and Computational Intelligence,
College of Mathematics and Computer Science, Hebei University,
Baoding 071002, China
mczjh@126.com

Abstract. A simple ordinal classification approach (SOCA) has been proposed by Frank and Hall. SOCA is a general method, any classification algorithm such as C4.5, k nearest neighbors (KNN) algorithm and extreme learning machine (ELM) etc. can be applied to this approach. We find that in SOCA only ordering information of decision attribute is used to classify objects but the ordering information of conditional attributes is not considered. Furthermore we experimentally find that ordering information of conditional attributes can also improve the generalization ability of the classification method. In this paper, we propose an improved ordinal classification methodology by employing the ordering information of both condition and decision attributes. In addition, we analyze the sensitivity of the SOCA on performance to the underlying classification algorithms, for instance, C4.5, KNN and ELM. A number of experiments are conducted and the experimental results show that the proposed method is feasible and effective.

Keywords: Ordinal classification · Monotonic classification · Decision tree · Rank mutual information

1 Introduction

General classification algorithms do not consider the ordering information including the conditional attributes and the decision attribute. Actually, the ordering information can make contribution to classification. Classification problems with ordering information are called ordinal classification problems, which are also named monotonic classification problems.

As early as 1989, David et al. [1] have studied the ordinal classification problems, and proposed an approach for learning and classification of monotonic ordinal concepts. In the next few years, David developed other two algorithms for monotonic classification problems [2, 3]. In literature [2], ordinal classification for multi-attribute decision making has been studied for discrete domains. In reference [3], an information-theoretic machine learning algorithm with monotonicity maintenance was proposed.

Since the pioneering work of David, the ordinal classification problems have been investigated by many machine learning researchers. Potharst and Bioch [4] proposed an order preserving tree-generation algorithm for multi-attribute classification problems with k linearly ordered classes and an algorithm for repairing non-monotonic decision trees. Frank and Hall [5] provide a general method

SOCA for classification of data sets with ordering information in decision attribute, which partition the k -class ordinal problem into $(k-1)$ -binary class problems. SOCA enables standard classification algorithms, such as, C4.5, KNN, and ELM etc. to exploit the ordering information. The key superiority of SOCA is that it does not require any modification of the underlying learning algorithms. However, SOCA ignores the ordering information in conditional attributes. Feelders and Pardoel [6] studied the pruning problem of monotone classification trees, and proposed a pruning algorithm which can improve the calculation efficiency of induction of trees. Cardoso and Ricardo [7] investigated the criteria for measuring the performance of ordinal classification, and based on confusion matrix, proposed a new metric. Wojciech and Slowinski [8] discussed the problem of nonparametric ordinal classification with monotonicity constraints and presented a statistical framework for classification with monotonicity constraints. The main contribution of [8] is that a statistical theory for ordinal classification with monotonicity constraints is developed. Hu et al. [9, 10] extended the ideas of ID3 algorithm to the monotonic classification, and investigated the corresponding heuristic, i.e., rank mutual information, and the corresponding tree generation algorithm.

Recently, a number of new problems related to ordinal classification have attracted the attention of many machine learning researchers. One example is that in [11, 12] Hu et al studied the feature selection problem for monotonic classification. Based on their proposed rank mutual information, a feature selection algorithm was developed. Another example is that, based on cluster and variability analyses, Lin [13] proposed a feature selection algorithm for ordinal multi-class classification problems.

In addition, the cost-sensitive based ordinal classification problems were studied in [14–16]. Moewes and Kruse [18] extended the ordinal classification to fuzzy scenario, and proposed a fuzzy ordinal classification method. Many applications of ordinal decision trees can be found from references. For example, Baccianella et al. in [17] applied the ordinal classification technique to text information mining and obtained a promising performance.

From the literature we do not find a study on the extension of SOCA by considering the ordering information of conditional attributes, aiming at an improvement of accuracy for ordinal classification. In this paper, we make an attempt to conduct such an improvement. By considering the ordering information both in conditional attributes and decision attribute, we develop an improved model of ordinal classification SOCA. We analyze the sensitivity of the SOCA on performance to the underlying classification algorithms, for instance, C4.5, KNN and ELM. A number of experiments are conducted and the experimental results show that the proposed method is feasible and effective. The experimental results also show that our proposed approach is not sensitive to the underlying algorithms.

The paper is organized as follows. Section 2 provides some necessary related preliminaries. Section 3 presents our improved model of SOCA by considering the ordering information of both conditional attributes and decision attribute. The experimental results and their corresponding analysis are listed in Section 4. Section 5 concludes this paper.

2 Preliminaries

2.1 Transform k -Class Ordinal Classification Problem to $(k - 1)$ -Binary Classification Problem

SOCA is a kind of classification methodologies which transform a k - class ordinal classification problem into $(k - 1)$ -binary classification problems. The main process is to convert the class attribute C^* with ordered values C_1, C_2, \dots, C_k into $k - 1$ binary attributes. The original dataset can be transformed into $k-1$ derived dataset. The derived datasets contain the same attribute values for each instance, and the derived binary class value is transformed from original class value by using the rule:

If $C(x) > C_i$ then $C - binary_i(x) = 1$; else $C - binary_i(x) = 0$. where x is a sample, $C(x)$ is the value of original class, and $C - binary_i(x)$ is the value of ith derived binary class.

Based on each derived dataset, we can get a binary classifier, then we can get $k-1$ binary classifiers.

In the process of testing, each sample is processed by each of the $k - 1$ binary classifiers and then the probabilities P_1, P_2, \dots, P_{k-1} are obtained, where P_i represents the decision value of sample x in i th binary classifier.

The probability of each of the k ordinal decision values is calculated using the following formulas.

$$P(C_1) = 1 - P_1 \tag{1}$$

$$P(C_i) = P_{i-1} - P_i, 1 < i < k \tag{2}$$

$$P(C_k) = P_{k-1} \tag{3}$$

The class with maximum probability is assigned to the sample.

2.2 Ranking Information Entropy and Ranking Mutual Information

For the ordinal classification problems with monotonicity constraints, based on rank mutual information, an ordinal decision tree algorithm (REMT) was proposed [9]. In comparison with the general ordinal classification model which assumes that condition attributes and decision attribute are all ordinal or only the decision is ordinal, the current problem assumes an additional monotonicity constraint, given two samples x and y , if $x \leq y$, then we have $f(x) \leq f(y)$.

Definition 1. Let $U = \{x_1, x_2, \dots, x_n\}$ be a set of objects, $B \subseteq A$ where A is a set of attributes. The upwards ranking entropy is defined as

$$RH_B^{\geq}(U) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\geq}|}{n} \tag{4}$$

Similarly the downwards ranking entropy of the set U is defined as

$$RH_B^{\leq}(U) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\leq}|}{n} \tag{5}$$

Definition 2. Let U be a set of objects described with a set of attributes A , $B \subseteq A$, $C \subseteq A$. The upwards ranking mutual information of B and C is defined as

$$RMI^{\geq}(B, C) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\geq}| \times |[x_i]_C^{\geq}|}{n \times |[x_i]_B^{\geq} \cap [x_i]_C^{\geq}|} \quad (6)$$

and downwards ranking mutual information of B and C is defined as

$$RMI^{\leq}(B, C) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\leq}| \times |[x_i]_C^{\leq}|}{n \times |[x_i]_B^{\leq} \cap [x_i]_C^{\leq}|} \quad (7)$$

3 The Improved Approach

In this section, we first conduct an analysis on the sensitivity of SOCA about its performance to the underlying classification algorithms, and then present the improved approach.

3.1 The Analysis of SOCA Sensitivity

Frank and Hall claim that SOCA is applicable in conjunction with any base learner that can output class probability estimates. The performance of SOCA may change with the change of underlying classifier, in this paper, we analysis the SOCA sensitivity by applying C4.5, KNN and ELM to SOCA respectively.

In C4.5, the value of leaf node is defined as the probability of samples whose decision value is 1 in current leaf node. Then, to a testing sample, we can get its class probability estimates.

In KNN, the decision value is defined as the probability of samples whose decision value is 1 in the k nearest neighbor. Due to KNN can output class probability estimates, we can apply SOCA to it.

In ELM, the decision value of a sample is defined as the degree of the sample belong to class 1. Then we transform the degree to a probability. In this way, ELM can be applied to SOCA.

3.2 Our Improved Approach

According to SOCA, we transform k -class ordinal classification problem with monotonicity constraint to $(k - 1)$ binary-class ordinal classification problems. Then we use REMT to solve the $(k - 1)$ binary-class ordinal classification problems. In other words, we can get $k - 1$ derived datasets based on SOCA, and then execute the process of building tree with REMT to each derived dataset. The details of the process are as follows.

1. In the process of building tree, in order to create nodes for the tree, we need to select a_{\max} and c_{\max} that satisfy $MaxRMI_{c_j} = RMI(a_i, D)$ by computing the maximal ranking mutual information between candidate attributes and decision attribute. a_i is the candidate attribute, c_j is attribute values of candidate a_i . Then

we set a_{\max} as the extended node and divide the node base on c_{\max} . However, if ranking mutual information of node attributes less than a specific value, or all of the samples are classified to the same class, we need to generate a leaf node base on the current samples. Because that the decision value is 0 or 1. The probability of the samples whose decision values are 1 is assigned to the current leaf node. For example: current leaf node contains 5 training samples, the decision value of 3 samples is 1, and the others are 0. Then we assign $3/5$ as the leaf node's decision value. Thus, k ordinal classification problem with monotonicity constraint can attain $k - 1$ binary trees through the above method.

2. In the process of searching tree, each sample is processed by each of the $k - 1$ binary tree and obtain the probabilities P_1, P_2, \dots, P_{k-1} (the decision value of current sample in $k - 1$ binary tree). The probability assigned to the k decision values (C_1, C_2, \dots, C_k) of the sample is calculated using the following formulas.

$$P(C_1) = 1 - P_1 \quad (8)$$

$$P(C_i) = P_{i-1} - P_i, 1 < i < k \quad (9)$$

$$P(C_k) = P_{k-1} \quad (10)$$

The class with the maximum probability is assigned to the sample.

4 Experimental Results

In order to verify the effectiveness of the proposed approach, we conduct some experiments with 10-fold cross validation on UCI datasets and artificial monotonicity constraint datasets. The basic information of the selected UCI datasets is listed in Table 1. The experimental environment is PC with 2.2GHz CPU and 8G memory, the operating system is Windows 7, MATLAB 7.1 is the experimental platform.

Table 1. The basic of the selected UCI datasets

datasets	#attributes	#classes	#samples
Abalone	8	29	4177
Cancer	10	2	341
Car	6	4	864
Mushroom	22	2	2820
Diabetes	8	2	576
German	20	2	500
Ionosphere	34	2	176
Sat	36	7	4290
Segment	19	7	1540
Servo	4	44	111
Sonar	60	2	150
Wave	21	3	3332

4.1 Sensitivity Analysis of SOCA

In order to analyze the sensitivity, we apply SOCA to C4.5, KNN and ELM respectively on UCI datasets. The average performance is depicted in Figure 1.

We use paired two sided t-test to test that if there is significance between the two results obtained with C4.5, KNN, ELM and with SOCA+C4.5, SOCA+KNN, SOCA+ELM respectively. The significance level is set to 1%. The results are given in Table 2. Through the experiments, we found the SOCA is not sensitive to the underlying classifier.

Table 2. The results of paired two sided t-test

results	C4.5	KNN	ELM
improvement	7(4)	4(1)	10(5)
degradation	2(0)	2(0)	2(2)

4.2 Performance of Our Proposed Approach

We conduct another experiment on an artificial data set to analyse the performance of the proposed approach. The artificial data set are generated with the following formula:

$$f(x_1, x_2) = x_1 + \frac{1}{2}(x_2^2 - x_1^2) \quad (11)$$

Where, x_1 and x_2 are two random variables which are independently drawn from $[0, 1]$. In order to generate ordered class labels, the resulting values were discretized into k intervals $[0, 1/k], (1/k, 2/k], \dots, (k-1/k, 1]$. Thus each interval contains approximately the same number of samples. The samples belonging to one of the intervals share the same decision value. Then we form a k -class monotonic classification task. In this experiment, we try $k = 2, 4, 6, 8, 10, 20$ and 30 respectively.

We first study the proposed approach on different numbers of classes. We generate a set of artificial datasets with 1000 samples and the numbers of classes vary from 2 to 30. Based on 5-fold cross validation technique, the experiment was repeated 100 times. The average performance is computed and given in Table 3; the curves of testing accuracy with number of classes are shown in Figure 2. The proposed method yields the better testing accuracy in all the cases except the case two classes are considered.

In addition, we also consider the influence of sample numbers on the performance of trained models. For artificial data of 1000 samples with 6, 10, 20, 30 classes respectively, we randomly draw training samples from a dataset. The size of training samples ranges from 50 to 200. In this process, we guarantee there is at least one representative sample from each class.

The rest samples are used in testing for estimating the performance of the trained models. To the different classes, the curves of testing accuracy with

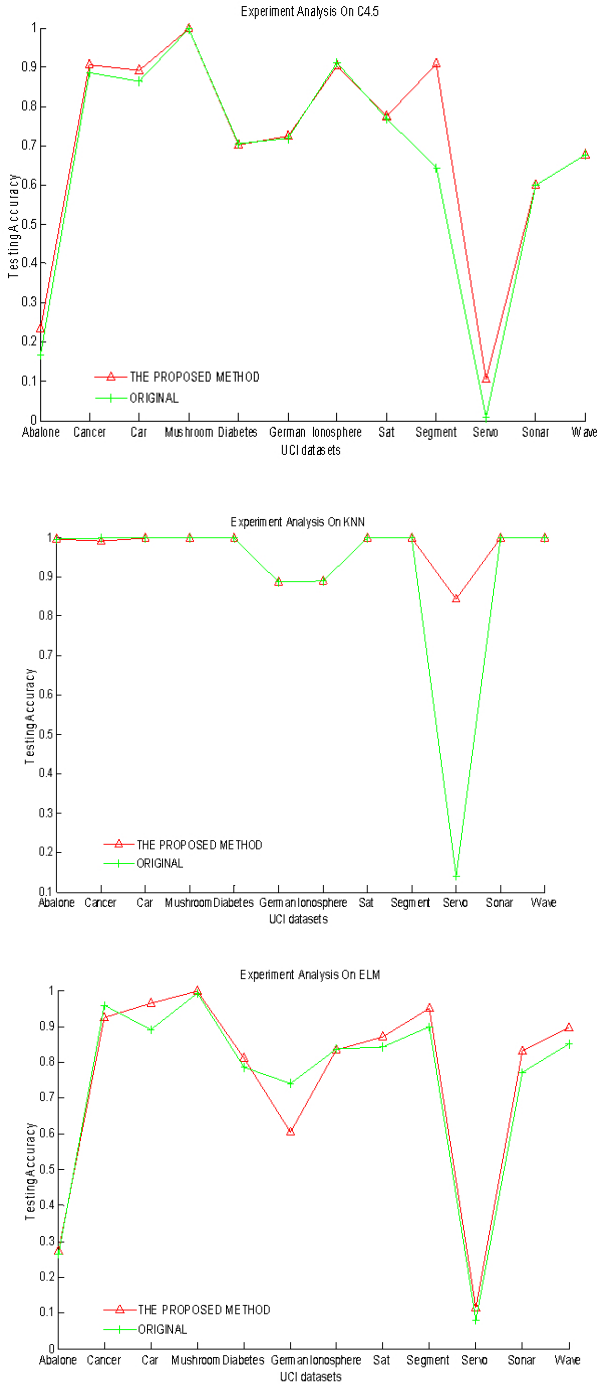
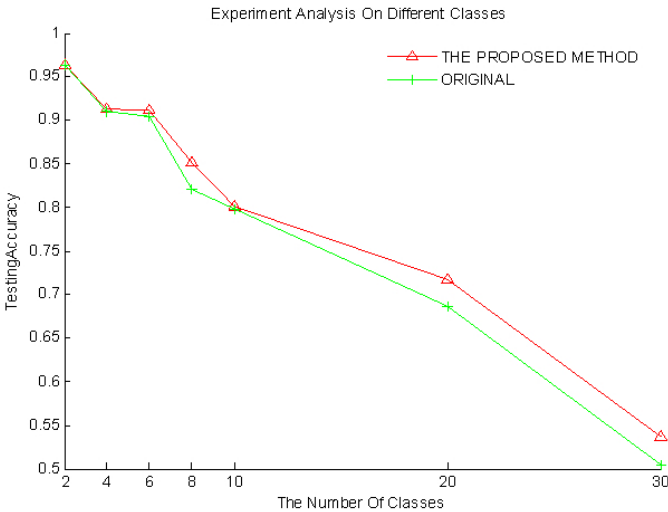


Fig. 1. The experimental results with different methods: C4.5, KNN and ELM

Table 3. The testing accuracy on artificial data with the number of class values vary from 2 to 30

Classes	REMT	The proposed method
2	0.9624	0.9626
4	0.9126	0.9096
6	0.9113	0.9043
8	0.8507	0.8210
10	0.8010	0.7984
20	0.7173	0.6859
30	0.5375	0.5054

**Fig. 2.** The testing accuracy on artificial data with different classes

number of training samples change are shown in Figure 2. We can see that the accuracy of the proposed method is higher than REMT, no matter how many training samples are used.

5 Conclusions

We extended the method SOCA, and proposed an improved ordinal classification approach in this paper. The ordering information both of conditional attributes and decision attribute are all be taken into consideration in the improved approach. We also analysed the sensitivity of the SOCA on performance to the underlying classification algorithms, and obtained the conclusion that SOCA is not sensitive to the underlying algorithms. In our future works, we will study

whether any ordinal classification algorithm whose outputs are posteriori probabilities can be used as the underlying algorithms, not only limited to the decision tree algorithm.

Acknowledgments. This research is supported by the national natural science foundation of China (61170040, 71371063), by the key scientific research foundation of education department of Hebei Province (ZD20131028), by the scientific research foundation of education department of Hebei Province (Z2012101), and by the natural science foundation of Hebei Province (F2013201110, F2013201220).

References

1. Ben-David, A., Sterling, L., Pao, Y.H.: Learning and classification of monotonic ordinal concepts. *Comp. Intell.* **5**, 45–49 (1989)
2. Ben-David, A.: Automatic generation of symbolic multiattribute ordinal knowledge-based DSSs: methodology and applications. *Decision Sciences* **23**, 1357–1372 (1992)
3. Ben-David, A.: Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning* **19**, 29–43 (1995)
4. Potharst, R., Bioch, J.C.: Decision trees for ordinal classification. *Intelligent Data Analysis* **4**, 99–111 (2000)
5. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Flach, P.A., De Raedt, L. (eds.) *ECML 2001. LNCS (LNAI)*, vol. 2167, pp. 145–156. Springer, Heidelberg (2001)
6. Feelders, A., Pardoel, M.: Pruning for monotone classification trees. In: Berthold, M., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) *IDA 2003. LNCS*, vol. 2810, pp. 1–12. Springer, Heidelberg (2003)
7. Cardoso, J.S., Ricardo, S.: Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence* **25**, 1173–1195 (2011)
8. Wojciech, K., Slowinski, R.: On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering* **25**, 2576–2589 (2013)
9. Hu, Q., Che, X., Zhang, L.: Rank entropy-based decision trees for monotonic classification. *IEEE Transactions on Knowledge and Data Engineering* **24**, 2052–2064 (2012)
10. Hu, Q., Guo, M., Da, R.: Information entropy for ordinal classification. *Science China-Information Sciences* **53**, 1188–1200 (2010)
11. Hu, Q., Pan, W., Song, Y.: Large-margin feature selection for monotonic classification. *Knowledge-Based Systems* **31**, 8–18 (2012)
12. Hu, Q., Pan, W., Zhang, L.: Feature Selection for Monotonic Classification. *IEEE Transactions on Fuzzy Systems* **20**, 69–81 (2012)
13. Lin, H.Y.: Feature selection based on cluster and variability analyses for ordinal multi-class classification problems. *Knowledge-Based Systems* **37**, 94–104 (2013)
14. Ruan, Y.X., Lin, H.T., Tsai, M.F.: Improving ranking performance with cost-sensitive ordinal classification via regression. *Information Retrieval* **17**, 1–20 (2014)
15. Lin, H.T., Li, L.: Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation* **24**, 1329–1367 (2012)

16. Kotsiantis, S.B., Pintelas, P.E.: A cost sensitive technique for ordinal classification problems. In: Vouros, G.A., Panayiotopoulos, T. (eds.) SETN 2004. LNCS (LNAI), vol. 3025, pp. 220–229. Springer, Heidelberg (2004)
17. Baccianella, S., Esuli, A., Sebastiani, F.: Using micro-documents for feature selection: The case of ordinal text classification. *Expert Systems with Applications* **40**, 4687–4696 (2013)
18. Moewes, C., Kruse, R.: Evolutionary fuzzy rules for ordinal binary classification with monotonicity constraints. *Soft Computing* **291**, 105–112 (2013)

Classification Based on Lower Integral and Extreme Learning Machine

Aixia Chen^(✉), Huimin Feng, and Zhen Guo

College of Mathematics and Computer Science, Hebei University, Baoding 071002, China
aixia_chen@163.com

Abstract. It is known that the non-linear integral has been generally used as an aggregation operator in classification problems, because it represents the potential interaction of a group of attributes. The lower integral is a type of non-linear integral with respect to non-additive set functions, which represents the minimum potential of efficiency for a group of attributes with interaction. Through solving a linear programming problem, the value of lower integral could be calculated. When we consider the lower integral as a classifier, the difficult step is the learning of the non-additive set function, which is used in lower integral. Then, the Extreme Learning Machine technique is applied to solve the problem and the ELM lower integral classifier is proposed in this paper. The implementations and performances of ELM lower integral classifier and single lower integral classifier are compared by experiments with six data sets.

Keywords: Non-linear integral · Lower integral · Extreme Learning Machine · Possibility distribution

1 Introduction

Non-additive set functions have described well the potential interaction of a group of attributes. Then, several types of non-linear integrals with respect to non-additive set functions have been defined. In [1], Sugeno generally defined the concept of fuzzy measure and Sugeno integral. In [2], the Choquet integral, which is the extension of the Lebesgue integral, was proposed. And the upper and lower integrals, which are two extreme specified indeterminate integrals, were given by Wang et al. in [3].

Many researchers have attempted to use non-linear integral as an aggregation operator in multi-attribute classification problems [4],[5],[6],[7],[8], and the results are very inspiring. In this kind of approaches, the decisions of different classifiers are fused into a final classification result by a non-linear integral with respect to a non-additive set function, which expresses the weights and the interactions of each classifier for a given class. In the nonlinear classification, Sugeno fuzzy integral and Choquet fuzzy integral have been used in [5],[6],[7]. The upper integral has been used in [13],[14] with the nonlinear classification, and the performance of upper integral classifier is competitive.

In this paper, we attempt to introduce ELM into the lower integral classifier. The lower integral is an extreme decomposition by which the least integration values can be obtained [3]. We know that the decision of non-additive set function is the difficult problem in fuzzy integral classification. As the efficiency and performance of Extreme Learning Machine, we will introduce ELM into the lower integral classifier. In our ELM lower integral classifier, the non-additive set functions are randomly generated and the huge task of learning non-additive set functions is avoided. However, due to the existence of weights β_j , we expect that ELM lower integral classifier has better performance than the single lower integral classifier.

In the following we briefly present the basic firstly. And then the ELM lower integral is proposed. In the end, the comparison of ELM lower integral classifier and single lower integral classifier are provided by experimenting with some data sets.

2 Fuzzy Measure and Integral

Because the feature spaces which we deal with are usually finite, the definitions of fuzzy measure and integrals will be presented in the restrictive case of finite spaces.

Fuzzy measures have been introduced by Sugeno [1].

Definition 1 Assume that X is a non-empty finite set and \wp is the power set of X . The fuzzy measure μ defined on the measurable space (X, \wp) is a set function $\mu : \wp \rightarrow [0, 1]$, which verifies the following axioms:

$$\mu(\Phi) = 0, \mu(X) = 1 \tag{1}$$

$$A \subseteq B \Rightarrow \mu(A) \leq \mu(B) \tag{2}$$

(X, \wp, μ) is said to be a fuzzy measure space.

Definition 2 [11] Assume that (X, \wp, μ) is a fuzzy measure space, and $X = \{x_1, x_2, \dots, x_n\}$. Assume that f is a measurable function from X to $[0, 1]$, and without loss of generality, $0 \leq f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq 1$, and $A_i = \{x_i, x_{i+1}, \dots, x_n\}$. The Sugeno integral and the Choquet integral of f with respect to the measure μ are defined as respectively

$$(S) \int f d\mu = \bigvee_{i=1}^n (f(x_i) \wedge \mu(A_i)) \tag{3}$$

$$(C) \int f d\mu = \sum_{i=1}^n (f(x_i) - f(x_{i-1})) \mu(A_i) \tag{4}$$

where $f(x_0) = 0$.

Definition 3 [3] Assume that $X = \{x_1, x_2, \dots, x_n\}$, and \wp is the power set of X . In this case, any function defined on X is measurable. The lower integral and the upper integral of f with respect to the set function μ can be defined as follows

$$(L) \int f d\mu = \inf \left\{ \sum_{j=1}^{2^n-1} \lambda_j \cdot \mu(E_j) \mid f = \sum_{j=1}^{2^n-1} \lambda_j \cdot \chi_{E_j}, \lambda_j \geq 0 \right\} \tag{5}$$

$$(U) \int f d\mu = \sup \left\{ \sum_{j=1}^{2^n-1} \lambda_j \cdot \mu(E_j) \mid f = \sum_{j=1}^{2^n-1} \lambda_j \cdot \chi_{E_j}, \lambda_j \geq 0 \right\} \tag{6}$$

where some λ_j may be zero and $E_j, j=1,2,\dots,2^n-1$ are subsets of X arranged in this way: the binary expression of $j, (j)_2 = b_n^{(j)} b_{n-1}^{(j)} \dots b_1^{(j)}$, is determined by

$$b_i^{(j)} = \begin{cases} 1 & \text{if } x_i \in E_j \\ 0 & \text{if } x_i \notin E_j \end{cases} \quad i = 1, 2, \dots, n. \tag{7}$$

where, $E_1 = \{x_1\}, E_2 = \{x_2\}, E_3 = \{x_1, x_2\}, E_4 = \{x_3\}, E_5 = \{x_1, x_3\}, E_6 = \{x_2, x_3\}, E_7 = \{x_1, x_2, x_3\}, \dots$

The evaluation of the upper and lower integral is essentially a linear programming problem, when the integrand f and the set function μ are known.

Table 1. The values of set function μ in example 1

Set	Value of μ
Φ	0
$\{x_1\}$	5
$\{x_2\}$	6
$\{x_1, x_2\}$	14
$\{x_3\}$	8
$\{x_1, x_3\}$	7
$\{x_2, x_3\}$	16
$\{x_1, x_2, x_3\}$	18

Example 1 [3] There are three workers x_1, x_2, x_3 working for $f(x_1)=10$ days, $f(x_2)=15$ days and $f(x_3)=7$ days, respectively, to manufacture a kind of products. Their efficiencies of working alone are 5, 6 and 8 products per day, respectively. Their joint efficiencies are not the simple sum of the corresponding efficiencies given above, but are listed in table 1.

It is equivalent to solve this following linear programming problem.

$$\begin{aligned}
 \min \quad & 5a_1 + 6a_2 + 14a_3 + 8a_4 + 7a_5 + 16a_6 + 18a_7 \\
 \text{s.t.} \quad & a_1 + a_3 + a_5 + a_7 = 10 \\
 & a_2 + a_3 + a_6 + a_7 = 15 \\
 & a_4 + a_5 + a_6 + a_7 = 7 \\
 & a_j \geq 0, j = 1, 2, \dots, 7
 \end{aligned} \tag{8}$$

By running the program of the lower integral, we obtain

$$(L) \int fd\mu = 154 \tag{9}$$

with $a_1 = 3, a_2 = 15, a_3 = 0, a_4 = 0, a_5 = 7, a_6 = 0, a_7 = 0$. It is the minimum value of the number of products made by these workers. The corresponding working schedule s_{x_1} and s_{x_2} work together for 7 days; then s_{x_1} works alone for 3 days and s_{x_2} works alone for 15 days.

As the length of the paper is limited, the properties of fuzzy integral are present in references [1],[2],[3],[4],[11].

3 Classification by Fuzzy Integral

3.1 Possibility Theory

Possibility theory, which is an extension of the theory of fuzzy sets and fuzzy logic, was proposed by L.A.Zadeh in 1978 [8]. It is an uncertainty theory and substitution of probability theory, which is used to deal with the incomplete information. Possibility theory has been used in a number of fields, such as interval analysis, database querying, data analysis, etc.

Definition 4 [9] Let X is a variable, which takes values in the universe of discourse U , and a value of X denoted by u . Informally, a possibility distribution Π_X is a fuzzy relation in U , which acts as an elastic constraint on the values that may be assumed by X , thus, if π_X is the membership function of Π_X , we have

$$Poss\{X = u\} = \pi_X(u), \quad u \in U \tag{10}$$

where the left-hand member denotes the possibility that X may take the value u and $\pi_X(u)$ is the grade of membership of u in Π_X . When it is used to characterize Π_X , the function $\pi_X : U \rightarrow [0,1]$ is referred to as a possibility distribution function.

Example 2 Let X is the age of a chairman. Assume that X is a real-valued variable and $55 \leq X \leq 70$. Then, the possibility distribution of X is the uniform distribution defined by

$$\pi_X(u) = \begin{cases} 1 & u \in [55,70] \\ 0 & \text{elsewhere} \end{cases} \tag{11}$$

3.2 Extreme Learning Machine

In the following we will briefly present Single hidden layer feedforward networks (SLFNs)[12].

For l arbitrary distinct sample (x_i, t_i) , where $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in R^n$, and $t_i = (t_{i1}, t_{i2}, \dots, t_{im})^T \in R^m$, standard single hidden layer feedforward networks with N hidden nodes and activation function $g(x)$ are mathematically modeled as

$$\sum_{j=1}^N \beta_j g(W_j X_i + b_j) = o_i \quad (12)$$

where W_j is the weight connecting the j th hidden node and the input nodes, $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ is the weight vector connecting the j th hidden node and the output nodes. b_j is the threshold of the j th hidden node. W_j and b_j are randomly generated, and β_j need to be learned.

Theorem 1. Any continuous target function $f(x)$ can be approximated by SLFNs with adjustable hidden nodes. In other words, given any small positive value ε , for SLFNs with enough number of hidden nodes (L) we have

$$|f_L(x) - f(x)| < \varepsilon \quad (13)$$

3.3 Learning Process

Let us consider the learning process of ELM lower integral classifiers now.

Assume that there are l_j samples $X_1^j, \dots, X_{l_j}^j$ in class C_j , and similarly for all classes C_1, \dots, C_m . We denote $l = \sum_{j=1}^m l_j$ as the total number of samples and use indices i, j, k to denote respectively a feature, a class and a sample.

Let X is an unknown sample. The function $\Phi(C_j)$ is said to be the discriminant function, which is described by the possibility distribution $\pi(C_j | X)$. Similarly, the function $\phi_i(C_j)$, is called the partial matching degree of X to class C_j with the attribute x_i , which is described by the possibility distribution $\pi(C_j | x_i)$. Using Cox's axioms for defining conditional measures, it is known that

$$\pi(C_j | x_i) = \pi(x_i | C_j) \quad \forall i, j \quad (14)$$

So, we should assign all $\pi(x_i | C_j)$ at first.

Learning of the possibility distributions

We will learn a known $\pi(x_i | C_j)$ as follows. And we use all the samples in class C_j to construct a "possibilistic histogram". At first, a classical histogram with h boxes

p_1, \dots, p_h , here $p_r = n_r/l_j$, with n_r the number of samples in box r , will be constructed from the samples, and the tightest possibility distribution π_1, \dots, π_h having the same shape as the histogram will be searched. Without loss of generality, assuming that $p_1 \geq \dots \geq p_h$, this is attained by $\pi_r = \sum_{s=r}^h p_s$. At last, we obtain the continuous shape of $\pi(x_i | C_j)$ by a linear interpolation of the values π_r .

Learning of lower integral networks

As we know, the determination of the set functions is the difficult problem of nonlinear integral classifiers. In this paper, the Extreme Learning Machine is applied in lower integral classifier to solve the problem. The scheme of ELM lower integral classifier could be briefly described as follows. D is the given training data, and T is the testing data.

(1) For each feature in class C_j samples, we determine the frequency histogram. With continuous feature i , determine h boxes and the corresponding frequencies p_i . With nominal feature i , consider each value of feature i as a box and the corresponding frequencies p_i .

(2)Rearrange the p_i , and determine the possibility distribution π_j^i of each feature. When the feature is continuous, the possibility distribution will be obtained by the linear interpolation.

(3) For l arbitrary distinct sample (x_i, t_i) , where $x_i = (x_{i1}, x_{i2}, \dots, x_{im})^T \in R^n$, and $t_i = (t_{i1}, t_{i2}, \dots, t_{im})^T \in R^m$, standard single hidden layer feedforward networks with N hidden nodes and activation function $g(x)$ are mathematically modeled as

$$\sum_{j=1}^N \beta_j g\left((L) \int f(x_i) d\mu_j\right) = o_i \tag{15}$$

where μ_j is the set function connecting the i th hidden node and the input nodes, $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ is the weight vector connecting the i th hidden node and the output nodes.

It is equivalent to minimize the cost function

$$E = \sum_{i=1}^l \left\| \sum_{j=1}^N \beta_j g\left((L) \int f(x_i) d\mu_j\right) - t_i \right\| \tag{16}$$

where, $\|\bullet\|$ denotes the norm of the vector.

In the end, one ELM lower integral network is produced for each class. In the method, the set functions are randomly generated and the huge task of learning set functions is avoided. However, due to the existence of weights β_j , the lower integral with the set functions can also show itself effectively and smoothly.

(4) Test the classifier on some data sets.

4 Test on Real Data

In order to investigate how well ELM lower integral classifier works, we conduct an experimental study on some UCI machine learning databases, which are extensively used in testing the performance of different kinds of classifiers. The information about data sets used in our experiments is listed in Table 2.

Table 2. Data used in experiment

Data Set	Number of examples	Number of classes	Number of attributes
Iris	150	3	5
Pima	768	2	9
Wine	178	3	14
Hayes	132	3	6
Ecoli	336	8	8
Tic-tac-toe	958	2	9

10-fold cross validation for 20 times worked at each data set in our experiments. Firstly, we construct the possibility histogram for each feature in the samples of class C_j . The histogram is a graphical data analysis method, which has summarized the distributional information of a variable. If a feature is continuous, the feature is divided into equal sized h boxes (the value of h between 7~15 is appropriate. If the size of samples of class C_j is not large enough, the value is lower). And p_1, \dots, p_h are the frequencies in each box. If the feature is nominal, each feature value is considered as a box. Without loss of generality, assuming that $p_1 \geq \dots \geq p_h$ and $\pi_r = \sum_{s=r}^h p_s$, then $\pi(x_i | C_j)$ is given by a linear interpolation of the values π_r .

The comparison of the accuracy between single lower integral and ELM lower integral are shown in table 3.

Then, the ELM lower integral classifier is trained, in which the set functions $\mu_j (j = 1, 2, \dots, N)$ are randomly generated and weights $\beta_j (j = 1, 2, \dots, N)$ are learned to minimize the cost function E . Due to the existence of weights β_j , the ELM lower integral classifier can also show itself effectively and smoothly.

Table 3. Comparison of the accuracy between single lower integral and ELM lower integral

Data Set	Lower Integral		ELM Lower Integral	
	Mean	Std Dev	Mean	Std Dev
Iris	0.9517	0.0088	0.9701	0.0079
Pima	0.7562	0.0122	0.7808	0.0094
Wine	0.9517	0.0095	0.9687	0.0071
Hayes	0.6420	0.0243	0.6671	0.0117
Ecoli	0.7495	0.0156	0.7711	0.0110
Tic-tac-toe	0.6501	0.0107	0.6861	0.0079

From Table 3, it can be seen that the ELM lower integral classifier works well for both nominal and continuous attributes. And a comparison of the accuracy between single lower integral and ELM lower integral is present. It is obvious that the performance of ELM lower integral classifier precede the single lower integral classifier. We know the decision of non-additive set function is the difficult problem of fuzzy integral classifiers. In our ELM lower integral classifier, the non-additive set functions are randomly generated. We can see that the computational complexity will be exponentially reduced. However, due to the existence of weights β_j , the ELM lower integral can also show itself effectively and smoothly.

5 Conclusion

In order to effectively use the information of each attribute, and motivated by the effectiveness of ELM, this paper we proposed ELM lower integral classifier. In this approach, the non-additive set functions are randomly generated and the huge task of learning set functions is avoided. So the learning speed of ELM lower integral classifier is very fast. As ELM lower integral classifier takes the weights of importance of individual attributes into account, it is able to model interaction between attributes in a flexible way. From the experimental results, we can see that the performance of ELM lower integral classifier is better than the single lower integral classifier. This paper has demonstrated that the effectiveness of the ELM lower integral classifier, but the relationship between the number of the boxes for continuous feature and the classification performance will be our future investigation.

Acknowledgements. This paper was supported by Project of Hebei Provincial Department of Education (No. Z2012101, No. QN20131055) and the Natural Science Foundation of Hebei University (No. 2010Q26).

References

1. Sugeno, M.: Theory of fuzzy integrals and its applications. Doct. Thesis. Tokyo Institute of Technology (1974)
2. Sugeno, M., Murofushi, T.: Choquet integral as an integral form for a general class of fuzzy measures. In: 2nd IFSA Congress, Tokyo, pp. 408–411 (1987)
3. Wang, Z., Li, W., Lee, K.H., Leung, K.S.: Lower integrals and upper integrals with respect to non-additive set functions. *Fuzzy Sets and Systems* **159**, 646–660 (2008)
4. Grabisch, M., Sugeno, M.: Multi-attribute classification using fuzzy integral. In: 1st IEEE Int. Conf. on Fuzzy Systems, San Diego, pp. 47–54 (March 8–12, 1992)
5. Tahani, H., Keller, J.M.: Information fusion in computer vision using fuzzy integral. *IEEE Trans. SMC* **20**(3), 733–741 (1990)
6. Grabisch, M., Nicolas, J.M.: Classification by fuzzy integral. Performance and Tests: *Fuzzy Sets and Systems* **65**, 255–271 (1994)
7. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research* **89**, 445–456 (1996)

8. Grabisch, M.: Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems* (Special Issue on 5th IFSA Congress) **69**, 279–298. (1995)
9. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* **1**, 3–28 (1978)
10. Dubois, D., Prade, H.: *Possibility Theory*. Plenum Press, New York (1988)
11. Wang, Z., Klir, G.J.: *Fuzzy measure theory*. New York, Plenum Press (1992)
12. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: Theory and applications. *Neurocomputing* **70**, 489–501 (2006)
13. Wang, X., Chen, A., Feng, H.: Upper integral network with extreme learning mechanism. *Neurocomputing* **74**, 2520–2525 (2011)
14. Chen, A., Liang, Z., Feng, H.: Classification Based on Upper Integral. In: *Proceedings of 2011 International Conference on Machine Learning and Cybernetics 2*, pp. 835–840 (2011)

User Input Classification for Chinese Question Answering System

Yongshuai Hou^(✉), Xiaolong Wang, Qingcai Chen, Man Li, and Cong Tan

Key Laboratory of Network-Oriented Intelligent Computation,
Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen 518055, China
houyongshuai@hitsz.edu.cn, wangxl@insun.hit.edu.cn,
qingcai.chen@gmail.com

Abstract. Restricted-domain question answering system gives high quality answer to questions within the domain, but gives no response or wrong answer for out of the domain questions. For normal users, the boundary of in-domain and out-domain is unclear. Most users often send out-domain inputs to the restricted-domain question answering system. In such cases, both no answer and wrong answer from the system will yield bad user experience. In this paper, an approach is proposed to solve the bad system response issue of the restricted-domain question answering system. Firstly, it uses a binary classifier to recognize in-domain user inputs and uses the restricted-domain question answering system to provide correct answer. Secondly, an user input taxonomy for out-domain user input is designed, and a classifier is trained to classify the out-domain user input based on the taxonomy. Finally, different response strategies are designed to respond to different classes of out-domain user inputs. Experiments and actual application on a restricted-domain question answering system shows that the proposed approach is effective to improve user experience.

Keywords: User input classification · Restricted-domain · Question classification · Question answering system

1 Introduction

In recent years, researches on question answering (QA) systems progress rapidly. Most QA systems focus on how to answer questions in a restricted domain or in a restricted sentence format. Those QA systems are called restricted-domain QA systems. This type of QA system answers user's questions with high accuracy in the system's restricted domains [1], [2]. Restricted-domain QA systems make information search easier for users. However, the system needs user's input questions which follow the system domain restriction and the specified sentence format. Users should know both the domain and the input format of the QA system. If user's questions are beyond the system limitation, the system returns no answer or wrong answer. For example, the system returns "Sorry, I don't understand your question. Please try another." if the user's question is beyond the system limitation.

However in the real world, most people are not professional users for the restricted-domain QA system. They even do not know the domain and the limitation of the QA

system when entering a query. Those users input questions following their daily expression habits and from all domains. Some users even use the QA system as a dialogue system. In these cases, the system cannot answer user's un-restricted inputs and can only respond with no answer or a predefined answer like "Sorry, I don't understand your question. Please try another". In some cases, the system will return wrong answers for out-domain user inputs which is even worse. Those cases reduce the accuracy of answers and give users bad user experience. So, it is necessary to distinguish users' inputs which meet the limitation of the restricted-domain QA system or not. Furthermore, designing diversify responses for those user inputs that do not meet the system limitation to improve user experience is also necessary.

There is a large amount of researches on question classification for QA systems. Widely used methods for question classification include rule-based methods, Naïve Bayes [3], Support Vector Machine (SVM) [4], [5] and KNN [6]. Usually, a question classification taxonomy is designed based on the question content, or a special question classification problem (like the UIUC taxonomy. It cannot be used in other question classification problems [7]. Fan Bu and Xingwei Zhu design a taxonomy suitable to general QA systems, and plan to use it to answer unrestricted-domain question by sending the question to different restricted-domain QA systems based on the question class [8]. Most researches on question classification focus on classifying questions by the domain, question format. Only very few of them focus on user inputs which are partly belonging to questions of the restricted-domain QA system.

In this paper, we propose a user input classification approach for restricted-domain QA systems to distinguish classes of users' inputs automatically. Firstly, a binary classifier is trained to judge whether the user input is a question in the required domain. If the question is in the domain, it is sent to the restricted-domain QA system. Secondly, we design a user input taxonomy including 14 classes for out of the system domain questions, and train a multi-class classifier to classify those user inputs. For the user input following the system restriction, we search answer from the restricted-domain QA system. For the user input which is out of the system domain, we design different response strategies for different user input classes. The processes for answering user inputs in the restricted-domain QA system with our method is shown in Figure 1.

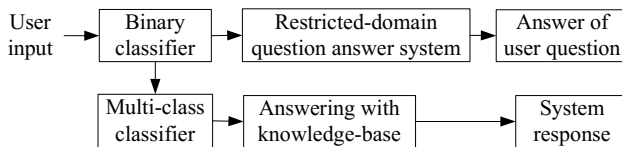


Fig. 1. Answering processes for user inputs

We define the *user input* as all cases of user inputs to the QA system, such as all questions, dialogue sentences, keywords for information search. The *in-domain user input* is defined as the user questions which meet the input limitation of the restricted-domain QA system. All other user inputs that do not follow the limitation of the QA system are defined as the *out-domain user input*.

2 User Input Taxonomy

Because restricted-domain QA systems can only give high quality answers to questions in their restricted domains, so it is necessary to distinguish the questions which a system can answer. Most users do not usually use the QA system. They are not familiar with the usage rules and domain restriction for the system. So those users cannot ensure their inputs are following the system restriction. Even more, some users think the QA system knows everything and input queries from all domains, or use the QA system as a dialogue system and send dialogue sentences to the system. Most QA systems cannot answer those out-domain user inputs. There are two ways in those systems to response to the out-domain user inputs: the first one was response nothing or response answers like “there is no answer, please try other questions”. This response way makes a bad user experience for the system; the second one is finding an answer which is closest in meaning to the user input. This way cannot get right answers in most case, and reduced answer accuracy of the system.

To improve the user experience and answer accuracy for restricted-domain QA system, we designed an answer strategy to response those out-domain user inputs. We firstly classified those out-domain user inputs, and designed different responses basing the user input classes. We designed a user input taxonomy for classifying user input for restricted-domain QA system, and the taxonomy is shown in figure 2.

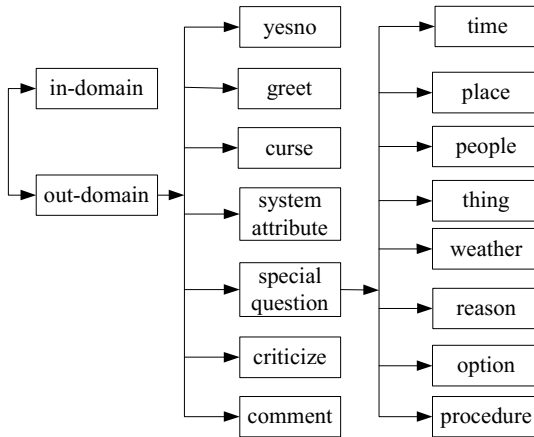


Fig. 2. User input taxonomy for restricted-domain QA system

3 User Input Classification

To answer user input separately basing on user input for restricted-domain QA system, we proposed an approach to classify and response user input for the restricted-domain QA system. The process for user input classification is shown in figure 1. First, we trained a binary classifier to recognize *in-domain* class question from user input and filter the *out-domain* class user input for the restricted-domain QA system. Second, we trained a classifier to classify the *out-domain* user input into 14 classes following the user input taxonomy we designed. Finally, we designed different response for different

class user input to ensure the restricted-domain QA system can send appropriate system response for most of user inputs, including *in-domain* and *out-domain*.

a) User input domain classification

A binary classifier was trained here to recognize user input domain for restricted-domain QA system. The features for the *in-domain* class user input to the restricted-domain QA system are clear and effective, and they are much suitable for the classifier based on rules. Here we used rule-based classifier to recognize the *in-domain* user input.

The restricted-domain QA system we used here is a financial-orient QA system (BIT Chinese Financial QA System)¹ we developed. So features we extracted for the binary classifier are financial-orient.

The features list we used to classify the domain of user input is shown as following:

- (1) Financial code: The code here includes stock code, fund code, bond code and so on. Regular expressions were used to match whether the user input contain financial code;
- (2) Financial named entity: Constructing a financial named entity dictionary includes name of stock, fund, bond, name of financial organize and so on, and judging whether the user input contain financial named entity in the dictionary;
- (3) Financial term: Constructing a financial term dictionary for those terms that are most used in financial domain, such as “interest on deposit”, “loan interest”, “exchange rate” and so on, and judging whether the user input contain financial term in the dictionary;
- (4) Special sentence structure: Using regular expressions to match whether user input is the special expression or sentence structure for financial domain;
- (5) Combination of the 4 features above.

b) Out-domain user input classification

The *out-domain* user input classification classified user input to 14 classes based on the taxonomy we designed. And machine learning methods were used on the classification. The features were selected based on Chinese short text classification methods [9] and the special characters for the QA system user input. Features list is shown as following:

- (1) Word feature: uni-gram, bi-gram, tri-gram of the user input;
- (2) Special word: dirty words, positive evaluation words, negative evaluation words, criticism words and so on;
- (3) Sentence templates and sentence combination patterns: Collecting and summarizing special sentence templates and sentence combination patterns from user input;
- (4) *Wh* word feature: *Wh* word, word before *Wh* word and word after *Wh* word, front word and tail word of user input;
- (5) Part of speech (POS): POS of word before and after *Wh* word, number of Verb, Noun and Adj. word in user input;
- (6) Semantic feature: semantic of word before and after *Wh* word;
- (7) Pronouns: whether there is personal pronouns in user input;
- (8) Syntactic structure: whether the syntactic structure of user input is complete;
- (9) Tail word: whether the tail word of user input is function word;

¹ bit.haitianyuan.com

4 Dataset and Experiment

a) Dataset

Most public dataset for QA system are just useful for question classification, and the classification taxonomy is aim on the question domains or syntactical structure. But there is few public user log for QA system as we know. And at same time, public dataset about classifying user input of QA system focus on the in-domain and out-domain case is also lacking.

In our experiments, we collected user logs from the financial-orient QA system (BIT Chinese Financial QA System) we developed. In this data set, financial-orient questions were used as *in-domain* questions and those didn't belong to financial domain input in the user log were used as *out-domain* user input. After removing repeated and invalid user inputs, we got 2196 different user input from the log of BIT QA system. Then we asked 3 persons to manually annotate classes of the user input and extracted the user input that at least 2 annotated results are same. Finally, we got 1669 manually annotated user input for experiment on classifying *in-domain* and *out-domain* user input, including 838 financial questions and 831 *out-domain* user inputs.

To train classifier for classifying the *out-domain* user inputs into 14 classes, using 831 user inputs as dataset is too small. So we extended the dataset scale by including the user inputs in AIML daily dialogue knowledge-base. After annotating and extracting, we got total 3024 different *out-domain* user inputs for classifier training. The detail user input distribution on the 14 classes is shown in table 1.

Table 1. Out-domain user input class distribution

Classes of out-domain user input	Number of user input
Yesno	303
Greet	215
Curse	218
System attribute	329
Criticize	181
Comment	149
Time	214
Place	157
People	188
Thing	152
Weather	204
Reason	285
Option	191
Procedure	238
Total	3024

b) Experiment

The user input classification performance is measured by precision (P), recall (R), F1 score ($F1$), calculate method are shown as formula (1), formula (2) and formula (3).

$$P = \frac{\sum_i \#correctly_classified_as_c_i}{\sum_i \#classified_as_c} \quad (1)$$

$$R = \frac{\sum_i \#correctly_classified_c_i}{\sum_i \#labeled_as_c_i} \quad (2)$$

$$F1 = \frac{2PR}{P + R} \quad (3)$$

There are rules to follow while classify financial-orient question. So we used rules based method to training the binary classifier. And the user input domain classification results with the binary classifier is shown in table 2.

Table 2. Experimental results of user input domain classification

	In-domain	Out-domain	Average
P	98.74%	94.52%	96.52%
R	94.22%	98.81%	96.52%
F1	96.43%	96.62%	96.52%

The classification precision for *in-domain* class user input is up to 98.74%, and the F1 score is 96.43%. The result shows that our binary classification method can ensure most user inputs send to the financial-orient QA system are financial questions. This is the basement for the financial-orient QA system to return right answers and appropriate system response. If there was no classifier to filter the out-domain user input for financial-orient QA system, the answer precision would decline and the user experience would become poor.

The binary classifying for user input can improve the answer precision. But to improve the system user experience, it must strengthen the response capacity to the *out-domain* class user input for the financial-orient QA system. We designed different response strategy to different user input class for the QA system.

With respect to the machine learning model, Naïve Bayes [3], Support Vector Machine (SVM) [4], [5] and KNN [6] are used in this paper. We compared the performance of the three machine learning methods on *out-domain* class user input classification and choose the best classifier.

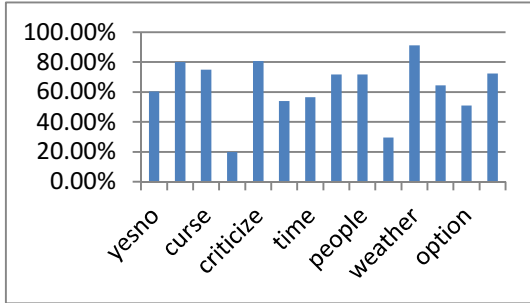
The three machine learning methods we used in this paper are from the toolkit RainBow², and all the parameters were set as default.

We used the 3024 annotated *out-domain* user inputs as experiment dataset. The dataset was divided into 3 parts, 2 parts were used as training data and 1 part was used as test data. We run the experiment with 3 folds cross validation. And the experiment results are shown as in table 3 and figure 3.

² <http://www.cs.cmu.edu/~mccallum/bow/rainbow/>

Table 3. Experimental results of out-domain user input classification with different algorithms

	Naïve Bayes	SVM	KNN
P	68.84%	65.47%	40.33%
R	63.50%	60.73%	46.76%
F1	61.99%	59.85%	43.27%

**Fig. 3.** F1 score of out-domain user input classification with Naïve Bayes

The results in table 3 show that the result of Naïve Bayes method is better than the other two machine learning methods. Basing on the experiment results we choose Naïve Bayes classifier to classify the *out-domain* user input for the financial-orient QA system. Figure 3 is the detail F1 score of each class in Naïve Bayes method results. Figure 3 shows that the F1 score of class *Greet*, *Curse*, *Criticize*, *Place*, *People*, *Weather* and *Procedure* are above 70%, and F1 score of other classes are low. Classification result of half number of classes is not satisfied to real requirement.

The accuracy of the whole classification results is low. The main reason is that the training data is insufficient. There are 14 classes but only 3024 instances to use, only 216 user inputs for each class in average. The training data is poor for each class. Another reason is that the feature for some classes is not discriminating to classification. For example, all the user inputs about system information are belong to class *System attribution*, but those user inputs has same features with other classes user input such as *Person* class. So some classes in the taxonomy should be adjusted.

In the practical application on the financial-orient QA system, to ensure the system give the right response, we designed some classification rules for some *out-domain* user input classes to avoid the influence of the low accuracy of classifier. With the user input, we first used the rules to classify some *out-domain* user inputs, then used the *out-domain* user input classifier to classify the user input that rules cannot cover. The real system we developed was shown in web <http://bit.haitianyuan.com>.

5 Conclusions

In this paper, we study the problem of how to classify the *in-domain* and *out-domain* user inputs and how to response the *out-domain* user input for the restricted-domain QA system. Firstly, we use a binary classifier to recognize *in-domain* user inputs for the restricted-domain QA system, and send *in-domain* user inputs to the system to ensure that the system provide accurate answers. Then, we design a 14 classes taxonomy to

classify *out-domain* user inputs and different response strategies for different user input classes. This step ensures that the restricted-domain QA system yields appropriate system responses to most of *out-domain* user inputs. Experimental results demonstrate that the classification of user input domain is effective, but more rules need to be added to the classification of the 14 classes *out-domain* user inputs before application in real systems to ensure the accuracy of the classification and accurate responses.

In our future work, we will collect and annotate more user input data for the classifier training on *out-domain* user inputs. We also plan to improve the 14 classes taxonomy by analyzing more user input data to make the class boundaries in the taxonomy more clear.

Acknowledgements. This paper is supported in part by grants: NSFCs (National Natural Science Foundation of China) (61173075 and 61272383), Scientific and Technological Research and Development Fund of Shenzhen (JC201005260175A), Strategic Emerging Industry Development Special Funds of Shenzhen (ZDSY20120613125401420 and JCYJ20120613151940045) and Key Basic Research Foundation of Shenzhen (JC201005260118A).

References

1. Hu, H., Ren, F., Kuroiwa, S., Zhang, S.: A question answering system on special domain and the implementation of speech interface. In: Gelbukh, A. (ed.) CICALing 2006. LNCS, vol. 3878, pp. 458–469. Springer, Heidelberg (2006)
2. Mollá, D., Vicedo, J.L.: Question answering in restricted domains: An overview. *Computational Linguistics* **33**(1), 41–61 (2007)
3. Lu, S., Chiang, D., Keh, H., et al.: Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values. *Knowledge-based Systems* **23**, 598–604 (2010)
4. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, July 28-August 1, pp. 26–32 (2003)
5. Pilászy, I.: Text categorization and support vector machine. In: The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence (2005)
6. He, X., Zhu, C., Zhao, T.: Research on short text classification for web forum. In: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, China, July 26-28, pp. 1052–1056 (2011)
7. Li, X., Roth, D.: Learning question classifiers: the role of semantic information. *Natural Language Engineering* **12**(3), 229–249 (2006)
8. Bu, F., Zhu, X., Hao, Y., et al.: Function-based question classification for general QA. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, Massachusetts, USA, October 09-11, pp. 1119–1128 (2010)
9. Fan, X., Hu, H.: A new model for chinese short-text classification considering feature extension. In: 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI), Sanya, China, October 23-24, pp. 7–11 (2010)

Fusion of Classifiers Based on a Novel 2-Stage Model

Tien Thanh Nguyen¹(✉), Alan Wee-Chung Liew¹, Minh Toan Tran²,
Thi Thu Thuy Nguyen³, and Mai Phuong Nguyen⁴

¹ School of Information and Communication Technology, Griffith University,
Nathan, QLD, Australia

{tienthanh.nguyen2, a.liew}@griffithuni.edu.au

² School of Applied Mathematics and Informatics,
Hanoi University of Science and Technology, Hanoi, Vietnam

toan.tranminh@hust.edu.vn

³ College of Economics, Hue University, Hue, Vietnam

nguyenthithuthuy@hce.edu.vn

⁴ College of Business, Massey University, Palmerston North, New Zealand
phuongnm0590@gmail.com

Abstract. The paper introduces a novel 2-Stage model for multi-classifier system. Instead of gathering posterior probabilities resulted from base classifiers into a single dataset called meta-data or Level1 data like in the original 2-Stage model, here we separate data in K Level1 matrices corresponding to the K base classifiers. These data matrices, in turn, are classified in sequence by a new classifier at the second stage to generate output of that new classifier called Level2 data. Next, Weight Matrix algorithm is proposed to combine Level2 data and produces prediction for unlabeled observations. Experimental results on CLEF2009 medical image database demonstrate the benefit of our model in comparison with several existing ensemble learning models.

Keywords: Multiple classifiers system · Combining classifiers · Classifier fusion · Ensemble method · Stacking algorithm

1 Introduction

Combining classifiers to achieve low error rate has received much attention in recent years. Many state-of-the art algorithms had been introduced and have been applied successfully to a variety of data sources. However, there is no one best combining algorithm. One algorithm may out-perform others on some datasets but underperform on others.

Duin [1] summarized six strategies to build a combining system with sound performance. The objective of these strategies is that “the more diverse the training set, base classifier and feature set, the better the performance of the ensemble system”. The idea behind this approach is to decrease the correlation between input training sets, between base classifiers, and between feature sets. The six strategies include (a) different initializations, (b) different parameter choices, (c) different architectures, (d) different classifiers, (e) different training sets, and (f) different feature sets.

ers by applying CA to matrices formed by transformed Level1 data and true label of these observations. On the other hand, Kittler [6] presented combining classifiers method based on fixed rules by introducing 6 fixed rules namely Sum, Product, Max, Min, Vote and Average.

Besides, a coarse 2-Stage approach was introduced by Benediktsson [10] in which two base classifiers were used, namely a Neural Network (ANN) classifier and a Gaussian Maximum Likelihood (GML) classifier in the first stage and another ANN classifier in the second stage. If both base classifiers agree, observation is classified to their agreed class. If they do not agree, the observation is rejected. All rejected observations are collected and classified with a second ANN in the second stage. This approach reached good results with several datasets although it is not general enough in most situations.

A 2-Stage combining model was published in [11] in which K different feature vectors extracted from an object is classified by K base classifiers respectively. However, extracting more than one feature vector is not always possible in most situations. In this case, Stacking-based approach is used. In the first stage, a dataset is classified by K base classifiers through Stacking to obtain the Level1 data. Next, the Level1 data is classified by a new classifier (denoted by C). This kind of model has two benefits. First, Level1 data usually has low dimensions compared with Level0 data, for instance, the feature vector of an image (Level0 data) might have more dimension than that of Level1 data. Moreover, Level1 data can be viewed as scaled result in $[0, 1]$ by K base classifiers while attributes in Level0 are frequently of diverse measurement unit and type. As a result, Level1 data could have more discrimination ability than Level0 data.

In this work, we propose a new 2-Stage model that has better performance than the original 2-Stage model of [11]. In Section 2, we introduce the architecture of our 2-Stage model and present one combining algorithm based on weights. Experimental evaluations are conducted on CLEF 2009 medical image database using the proposed model. Finally, we summarize and propose several possible further developments of the proposed model.

2 Proposed Model

In the original 2-Stage model, the posterior probability results from base classifiers are grouped into a single vector and this vector becomes the feature vector for C . In our approach, we separate the posterior probability results into K Level1 data corresponding to the K base classifiers, which are then classified by C in the second stage. After that, the K posterior probability results given by C are combined to give the final classification. Our idea is based on Strategy (e) in Section 1 where K Level1 data can be viewed as K different training sets. Our objective here is to increase the diversity so as to decrease the correlation between input training sets. The proposed model is illustrated in Figure 1.

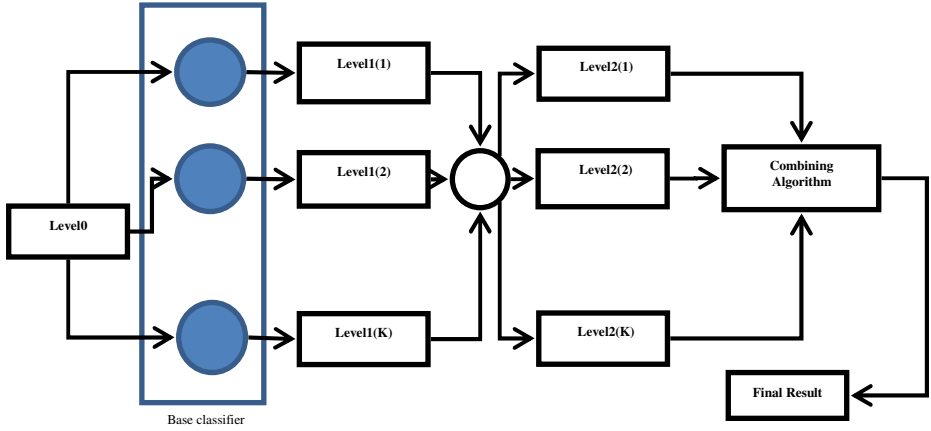


Fig. 1. The proposed 2-Stage model

In our model, Level1(k) of all observations from the k^{th} classifier is a $N \times M$ - posterior probability matrix $\{P_k(W_m | X_i)\}$ (3). After classified by C, Level2(k) is also a $N \times M$ - posterior probability matrix $\{P_k^C(W_m | X_i)\}$ $m = \overline{1, M}$ $i = \overline{1, N}$ (4).

$$Level1(k) := \begin{bmatrix} P_k(W_1 | X_1) & P_k(W_2 | X_1) & \dots & P_k(W_M | X_1) \\ P_k(W_1 | X_2) & P_k(W_2 | X_2) & \dots & P_k(W_M | X_2) \\ \dots & \dots & \dots & \dots \\ P_k(W_1 | X_N) & P_k(W_2 | X_N) & \dots & P_k(W_M | X_N) \end{bmatrix} \quad (3)$$

$$Level2(k) := \begin{bmatrix} P_k^C(W_1 | X_1) & P_k^C(W_2 | X_1) & \dots & P_k^C(W_M | X_1) \\ P_k^C(W_1 | X_2) & P_k^C(W_2 | X_2) & \dots & P_k^C(W_M | X_2) \\ \dots & \dots & \dots & \dots \\ P_k^C(W_1 | X_N) & P_k^C(W_2 | X_N) & \dots & P_k^C(W_M | X_N) \end{bmatrix} \quad (4)$$

while Level2 data of individual observation X is defined by:

$$Level2(X) := \begin{bmatrix} P_1^C(W_1 | X) & \dots & P_1^C(W_M | X) \\ \vdots & \ddots & \vdots \\ P_K^C(W_1 | X) & \dots & P_K^C(W_M | X) \end{bmatrix} \quad (5)$$

It is reasonable to assume that the contribution of each base classifier on combining result would not be the same since some base classifiers would be better suited to the classification problem and would contribute more to the combining result. Hence, we propose a combining algorithm where the weights of base classifiers on C are different. Our idea is to expand the Sum rule [6] by building weights as a

$K \times M$ matrix $\Psi = \{\omega_{km}\}^T$ in which k^{th} base classifier puts a weight ω_{km} on C corresponding to m^{th} class. We only impose a condition that all weights be non-negative $\omega_{km} > 0 \quad \forall k, m$. Performing combination task based on Ψ , we have M sums for X according to the M classes:

$$class\ m^{th} \leftarrow \sum_{k=1}^K \omega_{km} P_k^C(W_m | X) \quad m = \overline{1, M} \tag{6}$$

Algorithm 1. Generate K Level2 data

Input: original data (Level0), K base classifiers, classifier C.
 Output: K Level2 {Level2(k) | k=1, K}
 Step1: Use Stacking algorithm with K base classifiers to generate K posterior probability matrices corresponding to base classifiers {Level1(k) | k=1, K} (3)
 End
 Step2: For k = 1 to K
 Use Stacking algorithm on Level1(k) and classify by C to generate Level2(k) k=1, K (4)
 End

Here, m^{th} column vector of Ψ associated with m^{th} class is extracted: $W_m = \{\omega_{km}\}^T$ $k = \overline{1, K}$ for each $m = \overline{1, M}$. $N \times K$ Level2 posterior probability matrix of m^{th} class is given by:

$$P_m = \begin{pmatrix} P_1^C(W_m | X_1) & \dots & P_K^C(W_m | X_1) \\ \vdots & \ddots & \vdots \\ P_1^C(W_m | X_N) & \dots & P_K^C(W_m | X_N) \end{pmatrix} \tag{7}$$

Let the label vector be:

$$Y_m = \{y_m(X_i)\}^T \quad i = \overline{1, N} \tag{8}$$

where $y_m(X_i)$ is Crisp Label of X_i corresponding to m^{th} class.

$$y_m(X_i) = \begin{cases} 1 & \text{if } X_i \in W_m \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Our objective is to minimize the distance between sum (6) and true class label of all observation in training set. So from (7) and (8) we have a system of equations:

$$P_m W_m = Y_m \tag{10}$$

For $m = \overline{1, M}$, we have M system of equations in total. Solving each of them individually by non-negative regression we will have all column vectors of Ψ

During testing, we compute M sums based on Ψ :

$$\sum_{k=1}^K \omega_{km} P_k^C(W_m | X_{test}) \quad m = \overline{1, M} \tag{11}$$

and class label of unlabeled observation X_{test} is predicted by:

$$X_{test} \in W_J \text{ if } J = \arg \max_{m=1,M} \left(\sum_{k=1}^K \omega_{km} P_k^C(W_m | X_{test}) \right) \quad (12)$$

Algorithm 2. Weight matrix generation

Input: Level0, K base classifier, classifier C

Output: Weight matrix

Step1: Call Algorithm1 to generate K level2 data {Level2(k) | k=1...K}

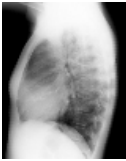

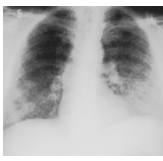

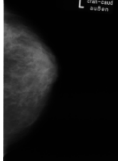





Step2: Build M matrixes P_m (7) and M label vectors Y_m (8) (9) $m=1,M$

Step3: Solve (10) to find Weight matrix Ψ

3 Experimental Results

We conducted experiment on CLEF 2009 database, a large set of medical image collected by Archen University. It includes 15,363 images allocated into 193 hierarchical categories. In our experiment we chose the first 7 classes and the entire 10 classes where each has different number of images (Table 1). First, we performed some pre-processing like histogram equation and then Histogram of Local Binary Pattern (HLBP) [15] is extracted as feature. Here Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Naïve Bayes were chosen as base classifiers and C is K Nearest Neighbor (with K set to 5, denoted as KNN(5)).

Table 1. Clef 2009 class information used in experiments

Image					
Description	Abdomen	Cervical	Chest	Facial cranium	Left Breast
Number of observation	80	81	80	80	80
Image					
Description	Left Elbow	Left Shoulder	Left Carpal Joint	Finger	Left Ankle Joint
Number of observation	69	80	80	66	80

We performed 10-fold cross validation and ran the testing 10 times so we had a total of 100 testing outcomes. Proposed model was compared with the original 2-Stage model, several classifiers combining algorithms, and KNN(5) on the original data

- Since our model is developed from the original 2-Stage model so it is important to have a comparison.
- Since our model is a combining classifiers model so it is necessary to compared with other well-known combining classifiers methods
- Since our model uses KNN in the second stage so it is important to compare with KNN applied on the original data to demonstrate effectiveness of Level1 data.

To assess statistical significance, we used paired t-test to compare two results (parameter α is set by 0.05). The results of the experiment are summarized in Tables 2 and 3.

Table 2. Classification error rate of well-known combining classifiers algorithms for clef2009

Combining classifiers methods	HLBP 7 classes		HLBP 10 classes	
	Mean	Variance	Mean	Variance
Sum rules	0.1125	1.70E-03	0.1525	1.47E-03
Product rules	0.1104	1.50E-03	0.1428	1.13E-03
Max rules	0.1291	1.54E-03	0.1572	1.63E-03
Min rules	0.1275	1.98E-03	0.1567	1.72E-03
Median rules	0.1191	1.77E-03	0.1656	1.25E-03
Majority Vote rules	0.1213	1.83E-03	0.1689	1.37E-03
Decision Template	0.1104	1.64E-03	0.1443	1.40E-03
SCANN	0.1187	1.36E-03	0.1528	1.62E-03

Table 3. Classification error rates of base classifiers and 2-stage model approaches for clef2009

	HLBP 7 classes		HLBP 10 classes	
	Mean	Variance	Mean	Variance
LDA	0.1318	1.96E-03	0.1684	1.24E-03
NaiveBayes	0.3320	3.16E-03	0.3694	3.10E-03
QDA	0.1418	1.77E-03	0.1732	1.78E-03
KNN(5)	0.2444	2.43E-03	0.2893	2.21E-03
Original 2-Stage Model	0.1200	1.62E-03	0.1459	1.40E-03
Proposed Model	0.1102	1.44E-03	0.1295	1.20E-03

The novel 2-Stage model outperforms fixed rules, posting 2 wins and 0 loss compared with Max, Min and Majority Vote rule and posting 1 win and 0 loss compared with Sum, Product and Median rule. Comparing with two well-known Stacking-based combining classifiers algorithms, our model achieves better results (1 win and 0 loss) (Table 4).

Table 4. Statistical test result between our combining algorithm and competitors for 7 or 10 classes

	Better	Competitive	Worse
Proposed Model vs. best result selected from base classifiers	2	0	0
Proposed Model vs. KNN(5)	2	0	0
Proposed Model vs. Original 2-Stage Model	2	0	0
Proposed Model vs. Sum rules	1	1	0
Proposed Model vs. Product rules	1	1	0
Proposed Model vs. Max rules	2	0	0
Proposed Model vs. Min rules	2	0	0
Proposed Model vs. Median rules	1	1	0
Proposed Model vs. Majority Vote rules	2	0	0
Proposed Model vs. Decision Template	1	1	0
Proposed Model vs. SCANN	1	1	0

Besides, our approach compared favorably with KNN(5) and the original 2-Stage model (Table 4). Our model achieves the lowest error (11% and 13%), followed by the original 2-Stage model (12% and 14.59%), best result selected from base classifiers (13.18% and 16.84%), and KNN(5) (24.44% and 28.93%) (Table 3). The benefit of Level1 data is demonstrated since our novel 2-Stage model using KNN(5) at second stage is better than KNN(5) on the original data. Experimental results also show the advantage of the proposed model compared with the original 2-Stage model as the consequence of using strategy (e) where K Level1 data are used as training sets for classifier at the second stage.

To sum up, our novel 2-Stage model outperforms KNN(5) on original data, the original 2-Stage mode, best result from base classifiers, as well as several well-known combining classifiers algorithms.

4 Conclusion

In this paper, we have introduced a novel 2-Stage model with weighted combining algorithm. Experimental results on CLEF2009 database demonstrated the benefits of the proposed model compared with KNN(5), the original 2-Stage model, and several well-known combining classifiers algorithms. In particular, combining based on weights in second stage outperforms all other benchmark algorithms. Our model is general in that we can use different base classifiers and classifier C, as long as these classifiers are diverse enough so as to decrease the correlation between K Level1 data.

In future work, we will study other combination strategies on Level2 data to further improve the performance of the proposed 2-Stage model. In addition, we will also look at applying classifier or feature selection methods to our 2-Stage model.

References

1. Duin, R.P.W.: The Combining Classifier: To Train or Not to Train? In: Proceedings of the 16th International Conference on Pattern Recognition, vol. 2, pp. 765–770 (2002)
2. Izenman, A.J.: *Modern Multivariate Statistical Techniques*, ch. 14. Springer, New York (2008)
3. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision Templates for Multi Classifier Fusion: An Experimental Comparison. *Pattern Recognition* **34**(2), 299–314 (2001)
4. Kuncheva, L.I.: A theoretical Study on Six Classifier Fusion Strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(2) (2002)
5. Ting, K.M., Witten, I.H.: Issues in Stacked Generation. *Journal of Artificial In Intelligence Research* **10**, 271–289 (1999)
6. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3) (1998)
7. Merz, C.: Using Correspondence Analysis to Combine Classifiers. *Machine Learning* **36**, 33–58 (1999)
8. Todorovski, L., Džeroski, S.: Combining Classifiers with Meta Decision Trees. *Machine Learning* **50**, 223–249 (2003)
9. Džeroski, S., Ženko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning* **54**, 255–273 (2004)
10. Benediktsson, J.A., Kanellopoulos, I.: Classification of Multisource and Hyperspectral Data Based on Decision Fusion. *IEEE Transactions on Geoscience and Remote Sensing* **37**(3) (May 1999)
11. Lepisto, L., Kunttu, I., Autio, J., Visa, A.: Classification of Non-Homogeneous Texture Images by Combining Classifier. In: Proceedings International Conference on Image Processing, vol. 1, pp. 981–984 (2003)
12. Sen, M.U., Erdogan, H.: Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters* **34**, 265–274 (2013)
13. Chen, Y.-S., Hung, Y.-P., Yen, T.-F., Fuh, C.-S.: Fast and versatile algorithm for nearest neighbor search based on a lower bound tree. *Pattern Recognition* **40**, 360–375 (2007)
14. Seeward, A.K.: How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness? In: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 554–561 (2002)
15. Ko, B.C., Kim, S.H., Nam, J.Y.: X-ray Image Classification Using Random Forests with Local Wavelet-Based CS-Local Binary Pattern. *J. Digital Imaging* **24**, 1141–1151 (2011)

Clustering and Kernel

Comparative Analysis of Density Estimation Based Kernel Regression

Junying Chen¹(✉) and Yulin He²

¹ College of Science, Agricultural University of Hebei,
Baoding 071001, Hebei, China
csjychen@126.com

² College of Mathematics and Computer Science,
Hebei University, Baoding 071002, Hebei, China
yulinhe@ieee.org

Abstract. The local linear kernel estimator (LLKE) is a typical kernel-type regression method which is a non-parametric method to estimate the conditional expectation of a random variable and the non-linear mapping from input to output. There are three commonly used LLKEs, i.e., the Nadaraya-Watson kernel estimator, the Priestley-Chao kernel estimator and the Gasser-Müller kernel estimator. Existing studies show that the performance of LLKE mainly depends on the selection of an important parameter, i.e. bandwidth h , when a special kernel function is employed. However, there is no comparative research conducted to study the effectiveness of different kernel functions. In this paper, we compare the performance of three aforementioned LLKEs based on 6 different kernel functions (i.e., Gaussian, uniform, Epanechnikov, biweight, triweight and cosine kernels) on their estimation error measured by the mean squared error (i.e., mse) and stability of method measured by the standard deviation of mse (i.e., std). Finally, we give guidelines for the selection of LLKE method and corresponding kernel function in practical applications.

Keywords: Bandwidth · Kernel regression · Local linear kernel estimator · Kernel function · Probability density estimation

1 Introduction

In statistics, the regression analysis [1–5] is a technique to establish the underlying function relationship between the observed inputs and target outputs. The Regression analysis has been widely used in prediction and forecasting, e.g., time series [6, 7] and economic activities [8, 9]. Let $S = \{(x_i, y_i) | x_i \in R, y_i \in R, i = 1, 2, \dots, N\}$ be the given dataset, where x_i and y_i are N observations of random variables X and Y . The objective of regression analysis is to find a mapping g from X to Y such that $y_i = g(x_i) + \varepsilon_i$, $i = 1, 2, \dots, N$, where $E(\varepsilon_i) = 0$ for each i . There are many well-known regression methods to find such mapping g . In this study, we focus on a simple but

very effective regression method, named the local linear kernel estimator (LLKE) which is also called the local polynomial kernel estimator (LPKE) [10].

The LLKE is a probability density function (p.d.f.) estimation based regression method. According to Eq. (1), the LLKE gives the estimation of a regression function g :

$$g(x) = \sum_{i=1}^N w_i y_i, \quad (1)$$

where, $w_i > 0$ is a function with respect to the input x and the weight of output y_i . $\sum_{i=1}^N w_i = 1$. The LLKE is to find the optimal weight set $\{w_1, w_2, \dots, w_N\}$ such that $g(x) = E(Y|X=x)$. Three mostly used methods to solve this problem w_i are the Nadaraya-Watson kernel estimator [11], the Priestley-Chao kernel estimator [12] and the Gasser-Müller kernel estimator [13].

The aforementioned methods use the p.d.f. estimation [14, 15] technique (e.g. the Parzen window method [14]) to determine the output weight. It is well-known that [16, 17] the selection of bandwidth h is very important to the performance of Parzen density estimation. The method for selecting an optimal bandwidth [15] is well studied in recent 20 years. However, there is no study to compare different kernel selection methods on the basis of kernel regression, e.g. the LLKE. Recently, Liu *et al.* [18, 19] compare the performance of different kernel functions in the framework of Flexible Naive Bayesian [15]. They found that the widely used Gaussian kernel is not always the best choice for the p.d.f. estimation. Motivated by studies of effectiveness of different kernel functions in [18] and [19], we compare the performance of three aforementioned LLKEs based on 6 different kernel functions (i.e., Gaussian, uniform, epanechnikov, biweight, triweight and cosine kernels) on their estimation error measured by the mean squared error (i.e., mse) [20] and method stability measured by the standard deviation of mse (i.e., std) [20] in this paper. There are three main contributions of this work. Firstly, we give a survey on different LLKE methods. Secondly, this is the first time that different kernel functions are employed in LLKE methods. Finally, extensive experiments are conducted to compare these algorithms and guidelines are derived for practical applications.

2 Three Local Linear Kernel Estimators

There are three main implementations for the local linear kernel estimator (LLKE), i.e., Nadaraya-Watson kernel estimator [11], Priestley-Chao kernel estimator [12] and Gasser-Müller kernel estimator [13]. Now, we give the detailed derivations about how to solve the output weight with these three methods:

2.1 Nadaraya-Watson Kernel Estimator-NWKE

NWKE [11] uses the following Eq. (2) to compute the output weight $w_i, i = 1, 2, \dots, N$ for y_i :

$$w_i = \frac{1}{Nh} \frac{K\left(\frac{x-x_i}{h}\right)}{\hat{p}(x)}, \quad (2)$$

Table 1. Five kernel functions

Kernel function $K(x)$	$ x \leq 1$	$ x > 1$
Uniform	$\frac{1}{2}$	0
Epanechnikov	$\frac{3(1-x^2)}{4}$	0
Biweight	$\frac{15(1-x^2)^2}{16}$	0
Triweight	$\frac{35(1-x^2)^3}{32}$	0
Cosine	$\frac{\pi \cos \frac{\pi}{2}x}{4}$	0

where, $\tilde{p}(x)$ is the estimated p.d.f. of random variable X based on the given observations x_1, x_2, \dots, x_N . According to the Parzen window method [14], we can get the expression of $\tilde{p}(x)$ as the following Eq. (3):

$$\tilde{p}(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right), \quad (3)$$

where, h is the bandwidth parameter.

By introducing Eqs. (2) and (3) into Eq. (1), we can get the regression function Eq. (4) with NWKE:

$$g_{\text{MWKE}}(x) = \frac{\sum_{i=1}^N [K(\frac{x-x_i}{h})y_i]}{\sum_{i=1}^N K(\frac{x-x_i}{h})}. \quad (4)$$

NWKE is a more natural method for the data usage in the regression analysis.

2.2 Priestley-Chao Kernel Estimator-PCKE

PCKE [12] determines the regression function as the following Eq. (5):

$$g_{\text{PCKE}}(x) = \frac{1}{h} \sum_{i=2}^N [(x_i - x_{i-1}) K(\frac{x-x_i}{h}) y_i], \quad (5)$$

where, the estimated p.d.f. $\tilde{p}(x)$ is $\tilde{p}(x_i) = \frac{1}{h(x_i - x_{i-1})}$. Without loss of generality, the input points are supposed to satisfy $x_1 < x_2 < \dots < x_N$. For PCKE, the output weight w_i is expressed as

$$w_i = \frac{(x_i - x_{i-1})}{h} K\left(\frac{x-x_i}{h}\right). \quad (6)$$

2.3 Gasser-Müller Kernel Estimator-GMKE

GMKE [13] achieves three advantages in comparison with PCKE: (1) no restriction to the positive kernels; (2) finding an asymptotically valid solution for the

boundary problem; and (3) smaller variance and mean square error. GMKE [13] determines the regression function as the following Eq. (7):

$$g_{\text{GMKE}}(x) = \frac{1}{h} \sum_{i=1}^{N-1} \left\{ \left[\int_{s_i}^{s_{i+1}} K\left(\frac{x-u}{h}\right) du \right] y_i \right\}, \quad (7)$$

where $s_i = \frac{x_{i-1} + x_i}{2}$, $i = 1, 3, \dots, N-1$ and $x_0 = 0$. Without loss of generality, the input points are also supposed to satisfy $x_1 < x_2 < \dots < x_N$ in GMKE. For GMKE, the output weight w_i can be expressed as

$$w_i = \frac{\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du}{h}. \quad (8)$$

3 Kernel Functions

A kernel $K(x)$ is a real-valued function which is non-negative, integrable and satisfies the following requirements:

$$\int_{-\infty}^{+\infty} K(x) = 1 \text{ and for } \forall x, K(x) = K(-x),$$

where, the first condition can ensure that the estimated function by using Parzen window [14] is a p.d.f.; and the second guarantees the estimated probability distribution is consistent with the true distribution of the samples used. In addition to the mostly used Gaussian kernel function formulated in Eq. (9):

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), x \in R, \quad (9)$$

there are also other five common kernels used in p.d.f estimation. Table 1 shows the detailed descriptions of these five kernels.

4 The Experimental Procedures and Results

In our experimental comparison, for the sake of running complexity, the rule-of-thumb method [16, 17] is used to determine the bandwidth h in Eqs. (4), (5), and (7):

$$h = \left(\frac{4}{3N}\right)^{\frac{1}{5}} \sigma, \quad (10)$$

where, $\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$ and $\mu = \frac{\sum_{i=1}^N x_i}{N}$ are variance and mean value of observations x_1, x_2, \dots, x_N respectively. We use an artificial dataset in which the input X and output Y satisfy the functional relationship listed Eq. (11) [16] and a practical dataset (i.e., the motorcycle dataset, p. 388 in [21]) as the experimental pool.

$$y = \sin(2\pi x) + 0.2\varepsilon, \quad (11)$$

where, $x \sim U(0, 1)$ and $\varepsilon \sim N(0, 1)$. Our experimental comparisons are divided into two parts: (1) giving a vivid presentation of regression performance of three aforementioned methods based on six different kernel functions. For GMKE, the solving of definite integral is used the standard Matlab implementations: *inline*¹ and *quad*². For every $y = \sin(2\pi x) + 0.2\varepsilon$ dataset, we randomly generate 200 points on which the comparison is conducted. The experimental results are listed in Figure 1. And, the fitting on the motorcycle dataset is in Figure 2. (2) providing the numerical comparisons concerning the estimated error and method stability. These two indexes are calculated as the following Eqs. (12) and (13) respectively.

- The performance of estimated error is measured by mean squared error between the real output y_i and the predicted output \tilde{y}_i :

$$mse = \frac{1}{N} \sum_{i=1}^N (\tilde{y}_i - y_i)^2. \quad (12)$$

- To measure the stability of regression method, we use the standard derivation of Q *mse*s, i.e.,

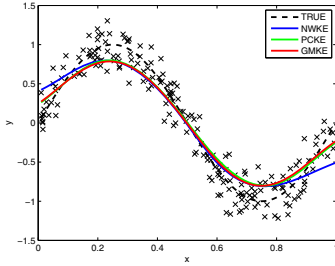
$$std = \sqrt{\frac{1}{Q} \sum_{i=1}^Q (mse_i - \mu)^2}, \mu = \frac{1}{Q} \sum_{i=1}^Q mse_i. \quad (13)$$

For $y = \sin(2\pi x) + 0.2\varepsilon$ regression, the final *mse* and *std* are respectively the average *mse* and standard derivation of $Q=10$ repetitions. The comparative results are summarized in Table 2 and Table 3. And, the estimated error on motorcycle dataset is in Table 4. From the experimental results in Tables 2, 3, and 4, we can summarize the following conclusions:

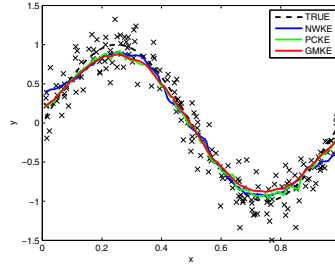
1. As demonstrated in [18] and [19], the mostly used Gaussian kernel is not the best choice for density estimation based regression analysis.
2. For the same LLKE, *triweight* kernel obtains the better fitting performance. This fact is also demonstrated by sub-figures 1-(e) and 2-(e). The main reason that *triweight* kernel achieves a more accurate estimation to the true validation function can be chalked up to its better stability. The comparison in Table 3 shows that *triweight* kernel has the smaller *std* and thus gets a more stable performance.
3. For the same kernel function, *GMKE* is the better implementation of LLKE. As emphasized in [13], *GMKE* has the smaller estimation variance and bias by finding the asymptotically valid solution for boundary problem. Figures 1 and 2 also present that the fitting of *GMKE* is closer to the true data distribution in comparison with other LLKEs. In addition, *GMKE* is a more stable LLKE as shown in Table 3.

¹ <http://www.mathworks.cn/cn/help/matlab/ref/inline.html>

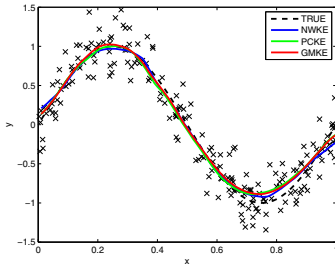
² <http://www.mathworks.cn/cn/help/matlab/ref/quad.html>



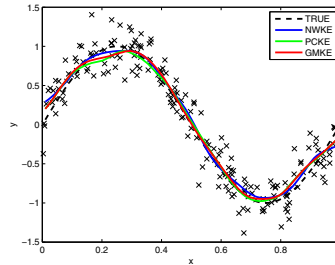
(a) The comparative results on Gaussian kernel



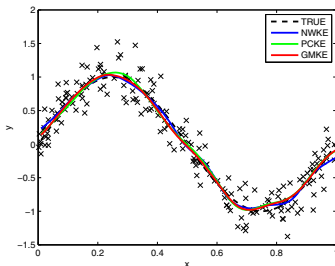
(b) The comparative results on Uniform kernel



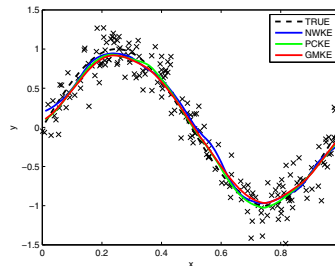
(c) The comparative results on Epanechnikov kernel



(d) The comparative results on Biweight kernel

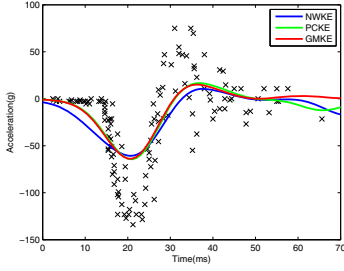


(e) The comparative results on Triweight kernel

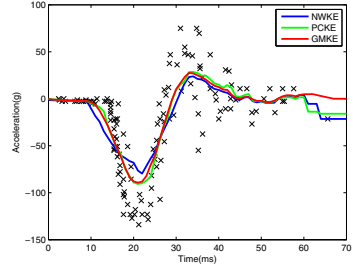


(f) The comparative results on Cosine kernel

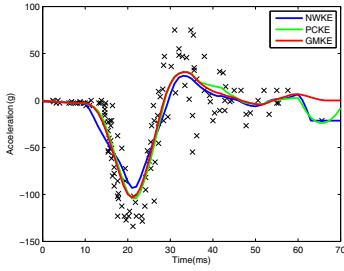
Fig. 1. The comparative results on $y = \sin(2\pi x) + 0.2\epsilon$ datasets



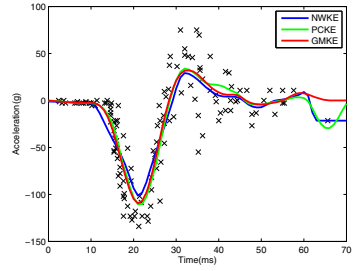
(a) The comparative results on Gaussian kernel



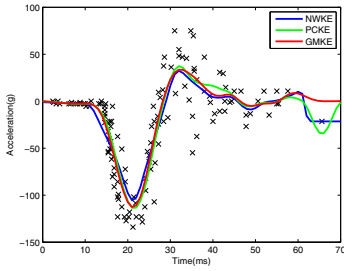
(b) The comparative results on Uniform kernel



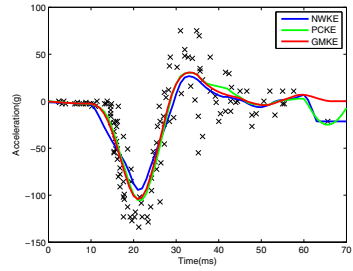
(c) The comparative results on Epanechnikov kernel



(d) The comparative results on Biweight kernel



(e) The comparative results on Triweight kernel



(f) The comparative results on Cosine kernel

Fig. 2. The comparative results on Motorcycle dataset

Table 2. The estimated error on artificial datasets

	GausK	UnifK	EpaK	BiK	TriK	CosK
NWKE	0.0824	0.0532	0.0397	0.0408	0.0365	0.0472
PCKE	0.0727	0.0498	0.0381	0.0432	0.0398	0.0439
GMKE	0.0719	0.0492	0.0368	0.0400	0.0353	0.0427

Table 3. The stability based on artificial datasets

	GausK	UnifK	EpaK	BiK	TriK	CosK
NWKE	0.0066	0.0069	0.0052	0.0066	0.0052	0.0061
PCKE	0.0076	0.0078	0.0068	0.0086	0.0043	0.0049
GMKE	0.0051	0.0066	0.0047	0.0072	0.0048	0.0035

Table 4. The estimated error ($\times 10^3$) on Motorcycle

	GausK	UnifK	EpaK	BiK	TriK	CosK
NWKE	1.1764	0.8802	0.7039	0.6260	0.5810	0.6883
PCKE	0.9938	0.6769	0.5501	0.5192	0.5093	0.5431
GMKE	1.0009	0.6560	0.5446	0.5088	0.4920	0.5370

5 Conclusions

In this paper, we compare the performance of three local linear kernel estimators based on the six commonly used kernel functions. Guidelines for practical applications are derived from extensive experiments.

References

1. Watson, G.S.: Smooth Regression Analysis. *Sankhyā: The Indian Journal of Statistics. Series A* **26**(4), 359–372 (1964)
2. Clevelanda, W.S., Devlin, S.J.: Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association* **83**(403), 596–610 (1988)
3. Chatterjee, S., Hadi, A.S.: *Regression Analysis by Example*. John Wiley & Sons (2006)
4. Montgomery, D.C., Peck, E.A., Vining, G.G.: *Introduction to Linear Regression Analysis*. John Wiley & Sons (2012)
5. Seber, G.A.F., Lee, A.J.: *Linear Regression Analysis*. John Wiley & Sons (2012)

6. Liu, Z.X., Liu, J.H.: Chaotic Time Series Multi-step Direct Prediction with Partial Least Squares Regression. *Journal of Systems Engineering and Electronics* **18**(3), 611–615 (2007)
7. Murtagh, F., Spagat, M., Restrepo, J.A.: Ultrametric Wavelet Regression of Multivariate Time Series: Application to Colombian Conflict Analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **41**(2), 254–263 (2011)
8. Zhang, X.R., Hu, L.Y., Wang, Z.S.: Multiple Kernel Support Vector Regression for Economic Forecasting. In: *Proceedings of ICMSE 2010*, pp. 129–134 (2010)
9. Zhang, S., Wu, Y.: The Application of Regression Analysis in Correlation Research of Economic Growth in Jilin Province and Regional Income Distribution Gap. In: *Proceedings of WKDD 2009*, pp. 500–503 (2009)
10. Fan, J.Q., Heckman, N.E., Wand, M.P.: Local Polynomial Kernel Regression for Generalized Linear Models and Quasi-Likelihood Functions. *Journal of the American Statistical Association* **90**(429), 141–150 (1995)
11. Watson, G.S.: Smooth Regression Analysis. *Sankhy: The Indian Journal of Statistics, Series A (1961–2002)* **26**(4), 359–372 (1964)
12. Priestley, M.B., Chao, M.T.: Non-Parametric Function Fitting. *Journal of the Royal Statistical Society Series B (Methodological)* **34**(3), 385–392 (1972)
13. Gasser, T., Müller, H.G.: Kernel estimation of regression functions. *Smoothing Techniques for Curve Estimation, Lecture Notes in Mathematics* **757**, 23–68 (1979)
14. Parzen, E.: On Estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics* **33**(3), 1065–1076 (1962)
15. Wang, X.Z., He, Y.L., Wang, D.D.: Non-Naive Bayesian Classifiers for Classification Problems with Continuous Attributes. *IEEE Transactions on Cybernetics* (2013), doi: 10.1109/TCYB.2013.2245891
16. Scott, D.W.: *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons, Inc. (1992)
17. Wand, M.P., Jones, M.C.: *Kernel Smoothing*. Chapman and Hall (1995)
18. Liu, J.N.K., He, Y.L., Wang, X.Z., Hu, Y.X.: A Comparative Study among Different Kernel Functions in Flexible Naive Bayesian Classification. In: *Proceedings of ICMLC 2011*, pp. 638–643 (2011)
19. Liu, J.N.K., He, Y.L., Wang, X.Z.: Improving Kernel Incapability by Equivalent Probability in Flexible Naive Bayesian. In: *Proceedings of FUZZ-IEEE 2012*, pp. 1–8 (2012)
20. Sun, Z.L., Au, K.F., Choi, T.M.M.: A Neuro-Fuzzy Inference System Through Integration of Fuzzy Logic and Extreme Learning Machines. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **37**(5), 1321–1331 (2007)
21. Härdle, W.: *Applied Nonparametric Regression*. Cambridge University Press (1994)

Thermal Power Units' Energy Consuming Speciality Analysis Based on Support Vector Regression (SVR)

Ming Zhao^(✉), Zhengbo Yan, and Liukun Zhou

Yunnan Electric Power Test and Research Institute Group Co. Ltd.,
Electric Power Research Institute, Kunming 650217, Yunnan Province, China
hdxuxiaogang@163.com

Abstract. There are some characteristics such as multi-borders, nonlinear time-variation of the thermal system of large coal-fired power units, the complex relationships between operating parameters and energy consumption, which affect the operation precision of thermal power units. According to rigorous theoretical analysis key operating parameters are identified and used to determine the standard coal consumption rate. On this basis, features are extracted and used as the inputs to SVR for training and testing. Energy consumption distribution model under full conditions of large coal-fired power units based on aforesaid method achieved a high precision.

Keywords: Energy consumption · Relationship · SVR · Data mining

1 Introduction

Thermal power units provide nearly 80 percent of the electricity in China, and this is a huge consumption of primary energy. More than half of the production of the coal is consumed on this in China. According to this Chinese energy situation, the proportion will not be greatly changed in the near future. The survey from the U.S. EPRI and the electric power industry show that, standard coal consumption [10] of the main generating units, even in the basic load, is higher than the designed value by about 30~40g/kWh. Therefore, energy-saving potential of power plants is huge, and this is closely related to the overall situation of Chinese energy consumption. The correct decision of the units' energy consumption characteristic is not only affecting the decisions for thermal power plants' energy optimization management [3] [13], but also for their practical values.

Plant thermal system is essentially a complex thermal system under multiple boundary conditions [8]. There are high-dimensional non-linear relationships among the operating parameters. The running state of the plant thermal system can be regarded as dynamic characteristics which meet uncontrollable system operating conditions (such as power generation load, ambient temperature, fuel characteristics, etc.), through the adjustment of the controllable parameters and the dynamic characteristics of equipment performance in a specific system environments. It ultimately

decides the operational status of the units, which show the corresponding energy consumption characteristics. For current large thermal power plants [15] which have complex thermal systems composed of many subsystems, in order to accurately describing the energy consumption characteristics under different operating conditions, and then achieving reasonable optimization to the controllable boundary condition parameters for establishing an optimizing strategies, the analysis of complex system is critical.

For having time [9] variations, nonlinear boundary conditions characteristics of the power plant systems, due to the complexity of the each subsystems' (such as the turbine, the boiler system, regenerative heating system, cold ends system) essential law for the dynamic processes, uncertainty of the physical structural changes (such as the scale of heat transfer equipment, blade variant, etc.) of the actual operation of equipment, complexity of the connections and their interaction rules [4] among the subsystems, the research route to reflect the rules of the relationship between the parameters of each state and then to analyze the units' energy consumption characteristics has been more restricted in practical applications through directly building the traditional mechanism model.

To avoid complex analysis, with the increasing levels of automation and powerful function of real-time database, mining association rules [6] [14] of operating status parameters from massive operating historical data becomes hot in these problems for its advantages such as target specific, units specific etc. Data mining [5] [12], information fusion [3] and other advanced algorithms based on artificial intelligence [7] are emerging.

Analysis of the units' energy consumption characteristics based on ϵ -SVR is proposed in the paper. The units' energy consumption characteristics model is designed under different loads and boundary conditions. Energy distribution model for large size coal-fired units of full-working-conditions is verified using real data.

2 Characterization of the Thermal Power Units' Energy Consumption

2.1 Analysis of the Thermal Systems' Boundary Conditions

Plant system is a complex thermal system [11] composed of many subsystems (equipment), and these are connected by specific way, which works under multi-boundary conditions, completes the process of energy conversion from thermal energy to mechanical energy and ultimately to electricity.

The composition of unit process and the scope of the study is shown in Figure 1.

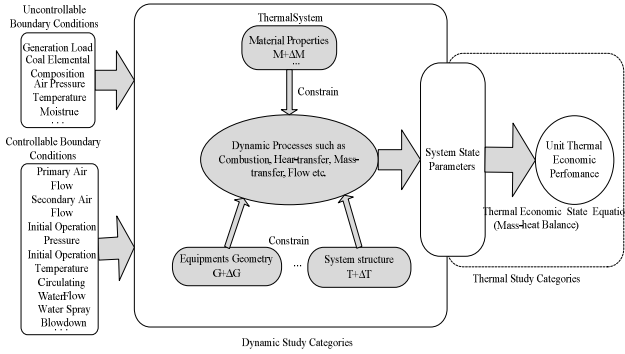


Fig. 1. Thermal unit process and scope of the study diagram

As shown above, thermal system with the certain device structures, material properties and system structures conducts the process of combustion, heat transfer, mass transfer and flow through each device inside, and this shows corresponding running state parameters in the constrained conditions such as uncontrollable boundary conditions (such as power generation load, the local meteorological conditions, coal quality, etc.), controllable boundary conditions (such as the turbine initial operating parameter, circulating water flow, boiler primary or second air flow, etc.) and system structures and device characteristics. These state parameters are ultimately expressed as the thermal and economic performance indicators of the unit’s thermal efficiency or power supply coal consumption rate.

2.2 Description of Unit Energy Consumption

The most fundamental indicators of energy performance of thermal power units usually expressed as power supply coal consumption rate b_{sn} as follow:

$$b_{sn} = \frac{123}{\eta_b \eta_i \eta_m \eta_g \eta_p (1 - \sum \xi_i)} \tag{1}$$

Where, $\eta_b, \eta_i, \eta_m, \eta_g, \eta_p, \sum \xi_i$ are respectively the boiler efficiency, cycle thermal efficiency, mechanical efficiency, generator efficiency, pipeline efficiency and auxiliaries electricity consumption rate.

To certain unit, equipment structure, material properties and system structure, even specific defects have been fixed. So b_{sn} can be expressed by function of system boundary conditions:

$$b_{sn} = f(N_g, T_{xrw}, D_w, P_0, T_0, T_{rh}, C_{coal}) \tag{2}$$

where, $N_g, T_{xrw}, D_w, P_0, T_0, T_{rh}, C_{coal}$ are respectively the load, circulating water inlet temperature (determined by the cooling tower performance and the ambient temperature), the circulating water flow, main steam pressure, main steam temperature, reheat steam temperature, coal characteristics.

3 Energy Consumption Distribution Model Based on ε -SVR

As a new technology in data mining, support vector machine (SVM) [1] was originally proposed in the 1990s by Vapnik, which is a new tool with the optimization method to solve machine learning problems. SVM is based on the statistical learning theory — VC dimension theory and structural risk minimization principle. Then the traditional empirical risk minimization principle has been changed. SVM has theoretical foundation and rigorous deduction process. Support vector regression (SVR) has advantages such as uniqueness of solution, global optimality, etc. It has unique advantage on the pattern recognition problems of small sample, nonlinear and high dimensionality, and so it was successfully applied in fields of industrial process control, non-linear classification, pattern recognition, and time-sequence prediction.

The concept of the feature space was proposed in SVM. The nonlinear problem in the original number field will be transformed into a linear problem in the feature space. And the kernel function was proposed. Then a linear problem in the feature space can be transformed into the nonlinear problem in the original number field. The feature space and the concrete form of nonlinear mapping are not mentioned, in order to obtain the best generalization ability.

For a given training set:

$$T = \{(x_1, y_1), (x_2, y_2) \cdots (x_l, y_l)\} \in (X, Y)^l,$$

$$x_i \in X = R^n, y_i \in Y = R, i = 1, 2, l.$$

Where x_i is the input variable, y_i is the corresponding target value, l is the number of samples. The target is to look for a real-valued function $f(x)$ within the scope of R^n , using $y = f(x)$ to infer the arbitrary pattern x corresponding y values.

For linear regression problem, it is to seek the classification hyperplane $f(x) = \omega \cdot x + b$, for all samples, that is:

$$|f(x_i) - y_i| \leq \varepsilon \quad (3)$$

Where $f(x)$ is as smooth as possible. It's so-called optimal regression hyperplane for the maximum interval regression classification hyperplane. Among them, ω is the adjustable weight vector, b is the bias. Regression hyperplane is called the best regression hyperplane when interval is $2/\|\omega\|_2$. The regression problem is a typical quadratic programming problems:

$$\min \frac{1}{2} \|\omega\|_2^2 \quad (4)$$

$$s.t. |f(x_i) - y_i| \leq \varepsilon (i = 1, \dots, l)$$

A nonlinear function $\Phi(\cdot)$ is used to map the input space R^n to a higher dimensional feature space Z . The optimal regression hyperplane $f(x)$ will be found in the feature space Z .

For the given training set:

$$T = \{(x_1, y_1), (x_2, y_2) \cdots (x_l, y_l)\} \in (X, Y)^l,$$

Where $x_i \in X = R^n$, $y_i \in Y = R$, $i = 1, 2, l$. As $\varepsilon > 0$, the ε -SVR algorithm can be expressed as the following optimization problem:

$$\begin{aligned} \min_{\omega, \xi, \xi^*, b} & \frac{1}{2} \omega^T \cdot \omega + C \sum_{i=1}^n (\xi + \xi^*) \\ \text{s.t.} & y_i - \omega^T \cdot \Phi(x_i) - b \leq \varepsilon + \xi_i \\ & \omega^T \cdot \Phi(x_i) + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi, \xi^* \geq 0 \end{aligned} \quad (5)$$

C is the penalty factor, which is used for regulating the smoothness and training accuracy of the function $f(x)$; $\xi^* = (\xi_1^*, \xi_2^*, \dots, \xi_l^*, \xi_l^*)^T$ is the slack variable, which is used for relaxation the constraints of SVM with hard boundary, ε is used to control the model fitting accuracy. The above optimization problem is solved by converting to dual problem through Lagrange method:

$$\begin{aligned} \min_{\alpha, \alpha^*} & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) - \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} & \sum_{i=1}^n (\alpha_i + \alpha_i^*) = 0 \\ & 0 \leq \alpha_i \leq C, \quad 0 \leq \alpha_i^* \leq C \end{aligned} \quad (6)$$

K is the kernel matrix, $\langle \cdot, \cdot \rangle$ is the vector inner product. $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$.

The model (4) has been solved to obtain the optimal solution $\bar{\alpha} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \dots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$, the resulting distribution model of the supply coal consumption:

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (7)$$

4 Example

The data in the paper was collected from a certain power plant unit No. 3, a total of 3250 data points from March 2006 to May 2006. The 2260 controllable sample points were obtained though quasi-steady-state test, error and redundant data elimination, and other data de-noising and cleaning process. The data was used as input of ε -SVR for training and testing. The unit energy consumption characteristics model under different loads and boundary conditions has been obtained with high precision validated by actual data. The results are shown in Figure 2.

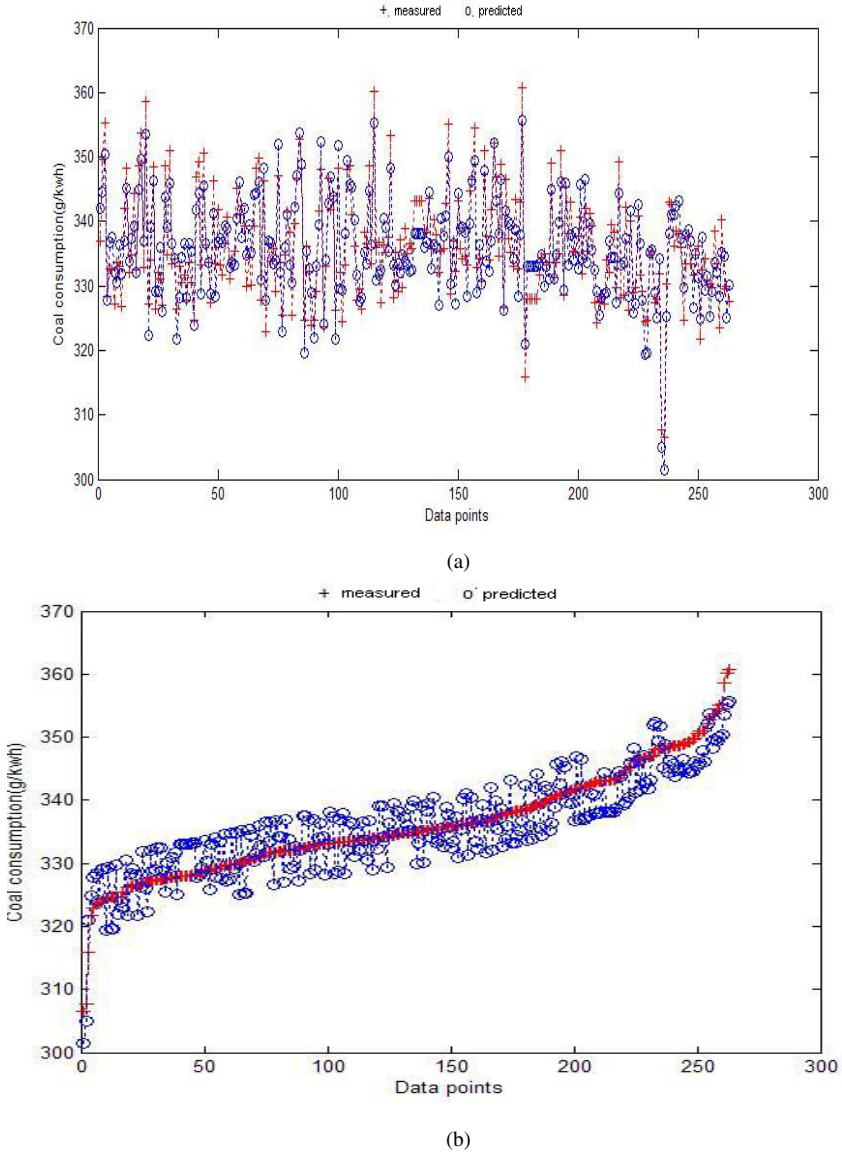


Fig. 2. Supply coal consumption distribution based on the ϵ -SVR

In the process, the radial basis function (RBF) [2] was chosen as the kernel function. The nonlinear sample can be processed by RBF in an effective manner.

Table 1. The features of the parameters σ and C in the model ($\varepsilon = 0.05$)

C	σ	Number of support vectors	The proportion	MSE (g)	Modeling time (s)
Inf	0.1	255	97.0%	3.8994e-002	22
Inf	1	75	28.5%	1.8442e-002	20
Inf	3	59	22.4%	1.7120e-002	20
Inf	5	70	26.6%	1.7449e-002	21
1	1	86	32.7%	2.4146e-002	20
5	1	76	28.9%	1.9186e-002	22
100	1	75	28.5%	1.8442e-002	23

As shown in Table 1, the value of σ and the C has a great significance for the model. The number of support vectors obtained by the model, modeling time and accuracy of the model were different with different values of the σ and C. In general, the size of the approximation error pipeline was controlled by the parameter ε in loss function, and then the number of support vectors and generalization ability were controlled by it. If the value was greater, then the accuracy was lower. Commonly, the range of ε was 0.0001~0.1. Penalty factor C is used for the compromise of controlling model complexity and approximation errors. It is found that the smaller the C, the greater the training error of the sample. It resulted in a larger risk for the structure of the model. Also, the larger the C, it will have a higher degree of data fitting. This will weaken the model generalization ability.

For RBF kernel function, the parameter σ has important impact on prediction accuracy. With σ increasing, the forecast performance of the model continuously improves, but when it rises to a certain value, over-fitting phenomenon appears, and the generalization ability degrades. The range was from 0.1 to 3.8 generally.

5 Conclusions

After a careful investigation on the many conditions about thermal power unit, a model that can prevent the phenomenon of "over-learning" is to fix C to 100 as the optimal parameter. When the support vectors are less, the minimum mean square error (MSE) and MSE average are 0.017g/kwh and 0.022g/kwh respectively. This shows that the proposed model has a high precision rate.

References

1. Jie, D., Wang, G., Han, P.: System identification based on support vector machines. *Computer Simulation* **21**(11), 39–41 (2004)
2. Wu, H., Liu, Y., et al.: Immune clustering-based radial basis function network model of short-term load forecasting. *Chinese Society for Electrical Engineering* **25**(16), 53–56 (2005)
3. Chen, J.H., Li, W., Sheng, D.: Line performance calculation of a thermal power units in data fusion. *Chinese Society for Electrical Engineering* **22**(5), 152–156 (2002)

4. Wang, H.: Thermal power units operating parameters of energy consumption sensitivity analysis. *Chinese Society for Electrical Engineering* **28**(29), 6–10 (2008)
5. Yan, T., Hu, Q., Bowen: Integration of rough sets and fuzzy clustering, continuous data knowledge discovery. *Electrical Engineering of* **24**(6), 205–210 (2004)
6. Hines, J.W., Uhrig, R.E., Wreast, D.J.: Use of Autoassociative Neural Network for Signal Validation. *Journal of Intelligent and Robotic Systems* **143** (1997)
7. Zhou, S.: Based on Wavelet Denoising and neural network theory of gas-solid circulating fluidized bed particle concentration prediction. *Petrochemical Technology* **32**(3), 224–229 (2003)
8. Chen, B., Su, H.N., Zhou, Y.: Steam Turbine Monitoring and Diagnosis System. *China Electrical Engineering* **24**(7), 253–256 (2004)
9. SI, F.-Q., Xu, Z.-G.: Based on the self-associative neural network and the measurement data since the calibration test method. *Chinese Society for. Electrical Engineering* **22**(6), 153–155 (2002)
10. Liu, F.: Power plant coal into the furnace element analysis and heat of the soft sensor real-time monitoring. *Electrical Engineering of* **25**(6), 139–146 (2005)
11. Zhang, Z., Tao, C., Xu, H., Hu, S.: Three high in the neural network ensemble forecasting model and its hair in the steam turbine unit state of repair. *Electrical Engineering of* **23**(9), 204–206 (2003)
12. Chen, H.: Combination forecasting method and its application, *University of Science and Technology* **9** (2002)
13. Yu, D., Hu, C., Xu, Z.-g.: Power plant performance analysis of sampled data reliability test method. *Power Engineering* **18**(2), 16–19, 74 (1998)
14. Li, J.: Optimization theory and applications of data mining-based power plant operation. *North China Electric Power University* (2006)
15. Toda, Hiromichi, Yamanaka: Planning and operation performance of 600 MW coal and oil dual-fired boiler for a supercritical sliding pressure operation. *Technical Review - Mitsubishi Heavy Industries* **22**(3), pp. 225–233 (October 1985)

Bandwidth Selection for Nadaraya-Watson Kernel Estimator Using Cross-Validation Based on Different Penalty Functions

Yumin Zhang^(✉)

Management Department, Hebei Finance University, Baoding 071051, Hebei, China
zhangyumin071051@126.com

Abstract. The traditional cross-validation usually selects an over-smoothing bandwidth for kernel regression. The penalty function based cross-validation (e.g., generalized cross-validation (CV_{GCV}), the Shibata's model selector (CV_S), the Akaike's information criterion (CV_{AIC}) and the Akaike's finite prediction error (CV_{FPE})) are introduced to relieve the problem of selecting over-smoothing bandwidth parameter by the traditional cross-validation for kernel regression problems. In this paper, we investigate the influence of these four different penalty functions on the cross-validation based bandwidth selection in the framework of a typical kernel regression method, i.e., the Nadaraya-Watson kernel estimator (NWKE). Firstly, we discuss the mathematical properties of these four penalty functions. Then, experiments are given to compare the performance of aforementioned cross-validation methods. Finally, we give guidelines for the selection of different penalty functions in practical applications.

Keywords: Cross-validation · Kernel regression · Nadaraya-Watson kernel estimator · Penalty function

1 Introduction

The kernel regression [1] is a non-parametric technique to construct the conditional expectation of a given random variable in statistics and is one of the most commonly used regression analysis methods. Its objective is to find a non-linear mapping between the input variable X and the output Y [2]. The conditional expectation of input X with respect to the output Y can be written as Eq. (1):

$$E(Y|X) = g(X), \quad (1)$$

where $g(X)$ is an unknown regression function that needs to be estimated [3]. Alternatively, we can rewrite Eq. (1) by the functional relation as follows:

$$Y = g(X) + \varepsilon, \quad E(\varepsilon) = 0. \quad (2)$$

Let $S = \{(x_i, y_i) | x_i \in R, y_i \in R, i = 1, 2, \dots, N\}$ be the given dataset, where x_i and y_i are N observations of random variables X and Y . According to Eq. (3), the kernel regression finds the estimation of function g [4]:

$$\tilde{g}(x) = \sum_{i=1}^N w_i y_i, \quad (3)$$

where $\tilde{g}(x)$ is the estimated regression function. w_i is a function with respect to the input x and denotes the weight of the output y_i ($w_i > 0$ and $\sum_{i=1}^N w_i = 1$). The target of kernel regression is to find an optimal weight set $\{w_1, w_2, \dots, w_N\}$ such that the estimated regression function $\tilde{g}(x)$ can approximate the true regression function $g(x)$ [5]. There are three commonly used methods to determine the output weight w_i : the Nadaraya-Watson kernel estimator [6], the Priestley-Chao kernel estimator [7] and the Gasser-Müller kernel estimator [8]. These three estimators are all implemented based on the Parzen window method [9].

It is well known that [10–12] the selection of smoothing parameter or bandwidth h is very important for the regression estimation performance of kernel regression methods. There are many sophisticated methods to find the optimal bandwidth, e.g., the cross-validation [13], penalizing functions [14], the plug-in [14] and the bootstrap methods [15]. None of them constantly outperforms the others. Therefore, we focus on studying the cross-validation based bandwidth selection which uses the leave-one-out strategy to determine the optimal bandwidth. In [14], some penalty functions are introduced to cross-validation to design an alternative calculation paradigm for cross-validation. In this paper, we introduce four different penalty functions based on cross-validation methods, i.e., the generalized cross-validation (CV_{GCV}) [16], the Shibata's model selector (CV_S) [17], the Akaike's information criterion (CV_{AIC}) [17] and the Akaike's finite prediction error (CV_{FPE}) [17]. We investigate the influence of these four different penalty functions on the cross-validation based bandwidth selection in the framework of the Nadaraya-Watson kernel estimator (NWKE). Firstly, mathematical properties of different penalty functions are discussed. Then, experiments are given to compare the performance of aforementioned cross-validation methods and we give guidelines for the selections of different penalty functions for future practical applications.

2 NWKE

NWKE [6] uses the following Eq. (4) to compute the output weight $w_i, i = 1, 2, \dots, N$ for y_i :

$$w_i = \frac{1}{Nh} \frac{K\left(\frac{x-x_i}{h}\right)}{\hat{p}(x)}, \quad (4)$$

where, $K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$ is the Gaussian kernel function and $\hat{p}(x)$ is the estimated p.d.f. of random variable X based on the given observations

x_1, x_2, \dots, x_N . According to the Parzen window method [9], we can get the expression of $\tilde{p}(x)$ as the following Eq. (5):

$$\tilde{p}(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right), \quad (5)$$

where, h is the bandwidth parameter.

By bringing Eqs. (4) and (5) into Eq. (3), we can get the regression function Eq. (6) with NW_{KRE} :

$$\tilde{g}_{\text{NWKE}}(x) = \frac{\sum_{i=1}^N [K(\frac{x-x_i}{h})y_i]}{\sum_{i=1}^N K(\frac{x-x_i}{h})}. \quad (6)$$

3 Cross-Validation Bandwidth Choice

NWKE is a more natural method for the data usage in the regression analysis. The regression performance of NWKE in Eq. (6) mainly depends on the selection of bandwidth parameter h . Cross-validation [13] is one of available bandwidth selection schemes, which uses the following formulas to determine the optimal bandwidth for NWKE:

$$h_{opt} = \arg \min_{h \in H} (CV(h)), \quad (7)$$

$$CV(h) = \sum_{i=1}^N \{y_i - \tilde{g}_{\text{NWKE}-i}(x_i)\}^2, \quad (8)$$

where, H is the domain of discourse of bandwidth h , $\tilde{g}_{\text{NWKE}-i}(x)$ is NWKE which is obtained without using the i -th instance (x_i, y_i) .

The penalty function based cross-validation calculates the optimal bandwidth according to the following Eqs. (9), (10) and (11):

$$h'_{opt} = \arg \min_{h \in H} (CV'(h)), \quad (9)$$

$$CV'(h) = \sum_{i=1}^N \{y_i - \tilde{g}_{\text{NWKE}-i}(x_i)\}^2 \pi(W(x_i)), \quad (10)$$

$$W(x) = \frac{K(0)}{\sum_{i=1}^N K(\frac{x-x_i}{h})}, \quad (11)$$

where, $\pi(u)$ is the penalty function. In this paper, we select four different penalty function based cross-validations as follows. Figure 1 gives the curve presentation of these four penalty functions.

- generalized cross-validation- CV_{GCv} :

$$\pi(u) = (1 - u)^{-2}; \quad (12)$$

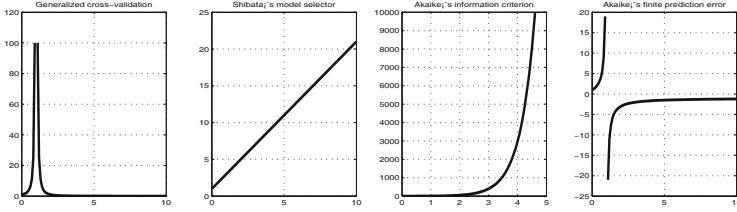


Fig. 1. 4 different penalty functions

- Shibata's model selector- CV_S :

$$\pi(u) = 1 + 2u; \quad (13)$$

- Akaike's information criterion- CV_{AIC} :

$$\pi(u) = \exp(2u); \quad (14)$$

- Akaike's finite prediction error- CV_{FPE} :

$$\pi(u) = \frac{1+u}{1-u}. \quad (15)$$

4 The Experimental Comparison Among Different Cross-Validations

In this section, we will conduct some experiments to compare the regression performances of NWKE with different bandwidth selection schemes, i.e., traditional cross-validation (CV), CV_{GCV} , CV_S , CV_{AIC} and CV_{FPE} . We compare the regression accuracy and method stability of above-mentioned five methods, and the regression accuracy and method stability are calculated as the following Eqs. (16) and (17) respectively:

- The fitting accuracy is measured by mean squared error between the real output y_i and the predicted output \tilde{y}_i :

$$mse = \frac{1}{N} \sum_{i=1}^N (\tilde{y}_i - y_i)^2. \quad (16)$$

- The stability is measured by the standard derivation of Q mse s, i.e.,

$$std = \sqrt{\frac{1}{Q} \sum_{i=1}^Q (mse_i - \mu)^2}, \mu = \frac{1}{Q} \sum_{i=1}^Q mse_i. \quad (17)$$

For the preparation of experimental datasets, we select six testing functions:

$$\begin{aligned} y_1 &= 1 - x + \exp \left[-200 (x - 0.5)^2 \right] + \varepsilon, \\ x &\sim \text{U}(0, 1), \varepsilon \sim \text{N}(0, 0.1); \end{aligned} \quad (18)$$

$$\begin{aligned} y_2 &= x + \frac{4 \exp(-2x^2)}{\sqrt{2\pi}} + \varepsilon, \\ x &\sim \text{U}(-3, 3), \varepsilon \sim \text{N}(0, 1); \end{aligned} \quad (19)$$

$$\begin{aligned} y_3 &= \sin \left[2\pi (1 - x)^2 \right] + x + 0.2\varepsilon, \\ x &\sim \text{U}(0, 1), \varepsilon \sim \text{N}(0, 1); \end{aligned} \quad (20)$$

$$\begin{aligned} y_4 &= x + 2 \sin(1.5x) + \varepsilon, \\ x &\sim \text{U}(0, 10), \varepsilon \sim \text{N}(0, 1); \end{aligned} \quad (21)$$

$$\begin{aligned} y_5 &= \left[\sin(2\pi x^3) \right]^3 + 0.2\varepsilon, \\ x &\sim \text{U}(0, 1), \varepsilon \sim \text{N}(0, 1); \end{aligned} \quad (22)$$

$$\begin{aligned} y_6 &= \sin(3\pi x) + 0.2\varepsilon, \\ x &\sim \text{U}(0, 1), \varepsilon \sim \text{N}(0, 1). \end{aligned} \quad (23)$$

We firstly give the descriptions concerning the relationship between the different CVs and bandwidths. Then, we compare the regression performances of NWKE with different optimal bandwidths h_{opt} . The comparative results on six testing functions are respectively summarized in Figures 2–7. For each testing function in this experiment, 200 data points are randomly generated. The minimal CVs and corresponding optimal bandwidths h_{opt} are listed in Table 1. We find the optimal bandwidth found by CV is smaller in comparison with other four CVs with different penalty functions. In other words, traditional CV in our comparisons is easier to obtain a rough bandwidth. Then, we give a more detailed comparison among aforementioned five methods (i.e., CV, CV_{GCV}, CV_S, CV_{AIC} and CV_{FPE}) in terms of regression accuracy and stability. For each testing function, the final *mse* and *std* are respectively the average and standard derivation of $Q=10$ repetitions. In every run, there are 200 data points which are generated randomly. We compare the performance of NWKE with different optimal bandwidths solved by five CVs respectively. The comparative results are summarized in Table 2. Through observing the experimental results, we can get the following conclusions:

- The traditional CV obtains the better regression accuracy and stability. It tells us that CV can select a more rough bandwidth for the smaller input (e.g., $x \in [0, 1]$, $x \in [-3, 3]$ or $x \in [0, 10]$ in the employed testing functions).
- CV_S achieves a better regression accuracy compared with CV_{GCV}, CV_{AIC} and CV_{FPE}. However, its stability is worse on the testing functions y_1 , y_2 , y_4 and y_5 .
- CV_{AIC} and CV_{FPE} obtain the comparative regression accuracy, but the stability of CV_{AIC} is better than CV_{FPE}.

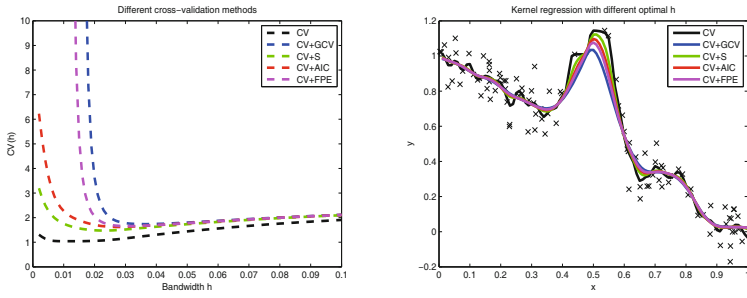


Fig. 2. The kernel regression on $y_1 = 1 - x + \exp[-200(x - 0.5)^2]$ dataset

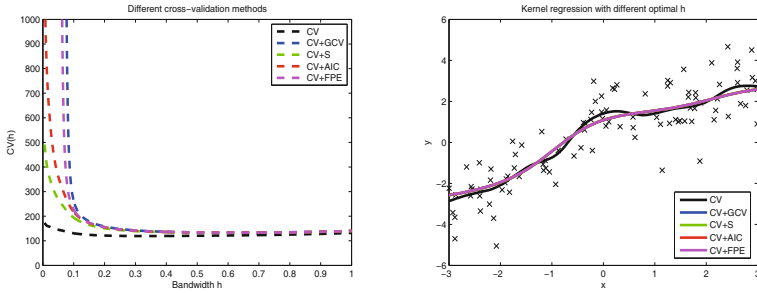


Fig. 3. The kernel regression on $y_2 = x + \frac{4 \exp(-2x^2)}{\sqrt{2\pi}}$ dataset

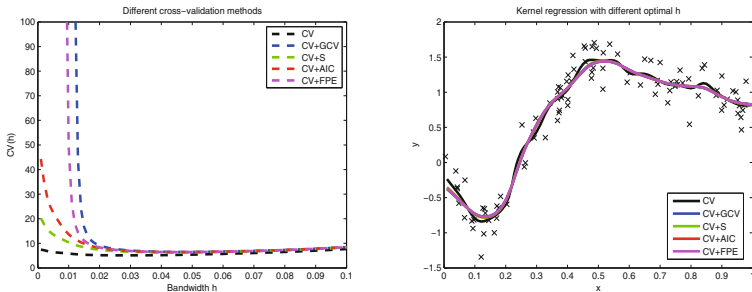


Fig. 4. The kernel regression on $y_3 = \sin[2\pi(1 - x)^2] + x$ dataset

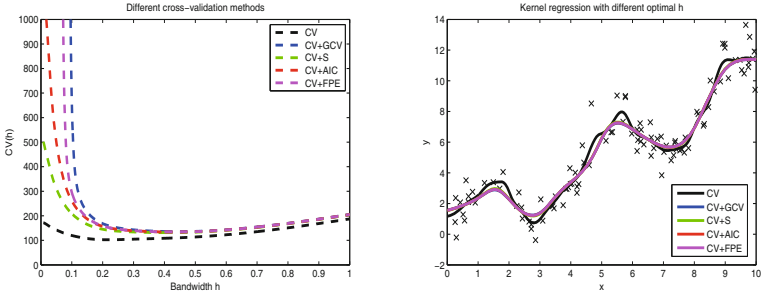


Fig. 5. The kernel regression on $y_4 = x + 2 \sin(1.5x)$ dataset

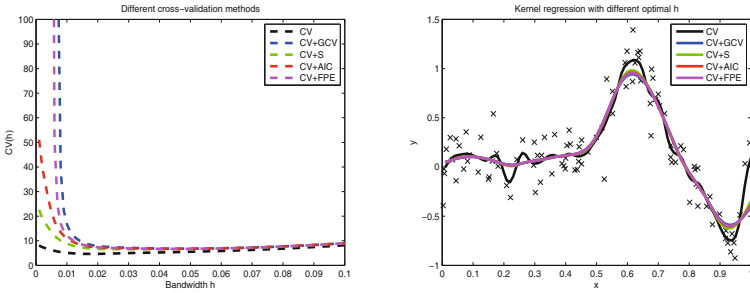


Fig. 6. The kernel regression on $y_5 = [\sin(2\pi x^3)]^3$ dataset

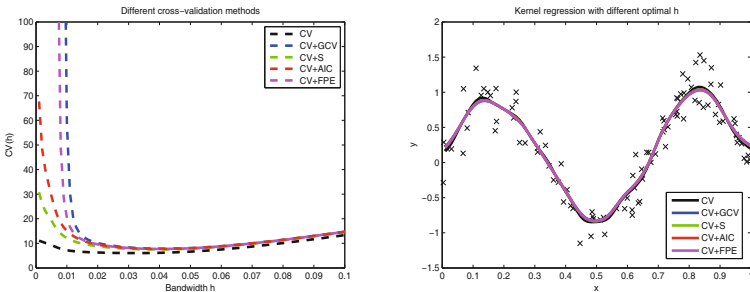


Fig. 7. The kernel regression on $y_6 = \sin(3\pi x)$ dataset

Table 1. The minimal CV and corresponding optimal bandwidth h_{opt}

CV methods	y_1		y_2		y_3		y_4		y_5		y_6	
	CV	h_{opt}	CV	h_{opt}	CV	h_{opt}	CV	h_{opt}	CV	h_{opt}	CV	h_{opt}
CV	1.042	0.012	119.364	0.330	5.115	0.029	102.400	0.218	4.685	0.021	6.224	0.029
CV+GCV	1.738	0.036	133.378	0.641	6.511	0.047	135.363	0.444	6.539	0.032	8.021	0.042
CV+S	1.474	0.023	132.633	0.620	6.313	0.043	131.399	0.414	6.176	0.029	7.749	0.038
CV+AIC	1.622	0.028	133.119	0.634	6.437	0.045	133.892	0.433	6.399	0.031	7.920	0.040
CV+FPE	1.656	0.031	133.126	0.634	6.443	0.046	133.994	0.434	6.413	0.031	7.928	0.041

Table 2. The regression performance of Nadaraya-Watson kernel estimator based on different CV methods

CV methods	Regression accuracy						Method stability					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
CV	0.0074	0.8217	0.0307	0.8549	0.0323	0.0276	0.0030	0.1898	0.0126	0.1705	0.0102	0.0113
CV+GCV	0.0097	0.9959	0.0489	1.1222	0.0488	0.0432	0.0039	0.1564	0.0064	0.2021	0.0165	0.0118
CV+S	0.0087	0.9693	0.0437	1.0434	0.0401	0.0342	0.0037	0.1551	0.0068	0.1930	0.0163	0.0110
CV+AIC	0.0093	0.9853	0.0461	1.0902	0.0448	0.0380	0.0037	0.1549	0.0072	0.1901	0.0168	0.0110
CV+FPE	0.0093	0.9866	0.0468	1.0966	0.0458	0.0398	0.0036	0.2873	0.0098	0.2678	0.0115	0.0131

5 Conclusions

In this paper, we investigate the influence of four different penalty functions on the cross-validation bandwidth selection in the framework of Nadaraya-Watson kernel regression estimator. The derived conclusions from our experiments give guidelines for the selection of different penalty functions for future applications.

References

- Harta, J.D., Wehrlya, T.E.: Kernel Regression Estimation Using Repeated Measurements Data. *Journal of the American Statistical Association* **81**(396), 1080–1088 (1986)
- Hart, J.D.: Kernel Regression Estimation With Time Series Errors. *Journal of the Royal Statistical Society, Series B: Methodological* **53**(1), 173–187 (1991)
- Herrmann, E.: Local Bandwidth Choice in Kernel Regression Estimation. *Journal of Computational and Graphical Statistics* **6**(1), 35–54 (1997)
- Dabo-Nianga, S., Rhomarib, N.: Kernel Regression Estimation in a Banach Space. *Journal of Statistical Planning and Inference* **139**(4), 1421–1434 (2009)
- Girarda, S., Guilloub, A., Stupfler, G.: Frontier Estimation with Kernel Regression on High Order Moments. *Journal of Multivariate Analysis* **116**, 172–189 (2013)
- Watson, G.S.: Smooth Regression Analysis. *Sankhy: The Indian Journal of Statistics, Series A (1961–2002)* **26**(4), 359–372 (1964)
- Priestley, M.B., Chao, M.T.: Non-Parametric Function Fitting. *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(3), 385–392 (1972)

8. Gasser, T., Müller, H.G.: Kernel estimation of regression functions. In: Gasser, T., Rosenblatt, M. (eds.) *Smoothing Techniques for Curve Estimation*. Lecture Notes in Mathematics, vol. 757, pp. 23–68. Springer, Heidelberg (1979)
9. Parzen, E.: On Estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics* **33**(3), 1065–1076 (1962)
10. Scott, D.W.: *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons, Inc. (1992)
11. Wand, M.P., Jones, M.C.: *Kernel Smoothing*. Chapman and Hall (1995)
12. Wang, X.Z., He, Y.L., Wang, D.D.: Non-Naive Bayesian Classifiers for Classification Problems with Continuous Attributes. *IEEE Transactions on Cybernetics* (2013), doi:10.1109/TCYB.2013.2245891
13. Leung, D.H.Y.: Cross-Validation in Nonparametric Regression With Outliers. *The Annals of Statistics* **33**(5), 2291–2310 (2005)
14. Härdle, W.: *Applied Nonparametric Regression*. Cambridge University Press (1994)
15. Härdle, W., Marron, J.S.: Bootstrap Simultaneous Error Bars for Nonparametric Regression. *The Annals of Statistics* **19**(2), 778–796 (1991)
16. Golub, G.H., Heath, M., Wahba, G.: Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter **21**(2), 215–223 (1979)
17. Wechsler, H., Duric, Z., Li, F.Y., et al.: Motion estimation using statistical learning theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(4), 466–478 (2004)

A Hough Transform-Based Biclustering Algorithm for Gene Expression Data

Cuong To, Tien Thanh Nguyen, and Alan Wee-Chung Liew^(✉)

School of Information and Communication Technology, Griffith University,
Logan, Australia
{a.liew,c.chieuto}@griffith.edu.au,
tienthanh.nguyen2@griffithuni.edu.au

Abstract. In pattern classification, when the feature space is of high dimensionality or patterns are “similar” on a subset of features only, the traditional clustering methods do not show good performance. Biclustering is a class of methods that simultaneously carry out grouping on two dimensions and has many applications to different fields, especially gene expression data analysis. Because of simultaneous classification on both rows and columns of a data matrix, the biclustering problem is inherently intractable and computationally complex. One of the most complex models in biclustering problem is linear coherent model. Several biclustering algorithms based on this model have been proposed in recent years. However, none of them is able to perfectly recognize all linear patterns in a bicluster. In this work, we propose a novel algorithm based on Hough transform that can find all linear coherent patterns. In the sequel we apply it to gene expression data.

Keywords: Biclustering · Linear coherent patterns · Additive and multiplicative models · Hough transform

1 Introduction

DNA microarray technologies allow us to measure expression levels for thousands of genes in various biological conditions. The raw data of a microarray experiment is an image which is then converted into a numeric matrix called gene expression matrix. Each row and column of a gene expression matrix presents gene and condition, respectively. We already knew that many genes have more than one function, and a group of genes can show similar expression under several conditions but not the others. Therefore, traditional clustering methods do not perform well in these cases. Biclustering performs simultaneous grouping on two dimensions and is able to find subset of genes that show similar expression behavior over a subset of conditions.

Several models [1, 2] have been proposed in biclustering such as constant (values, rows, columns) models, additive model, multiplicative model, and linear model. For the constant models [3, 4], all elements of a sub-matrix are a constant value. In the additive [5, 6] or multiplicative model [7, 8, 9], one column is obtained by adding a value to other column or is a factor multiple of other column. The general linear model,

first proposed in [27], [11], represents the relationship between two columns as a linear equation and it can be considered the general form of the models mentioned above. So the complexity of the general linear model is also the highest. Several algorithms based on the general linear model have been proposed recently [2], [10, 11, 12, 13], [27]. Specifically, Gan et al. [27], [11] were the first to formulate the biclustering problem as the detection of hyperplanes in high-dimensional space, and they proposed to apply Hough transform to find biclusters. The attractiveness of this formulation is that it is theoretically possible to detect all biclusters of the general linear type in a dataset as Hough transform does not depend on iterative optimization. The major weakness, however, is the extremely high computation cost when Hough transform is performed in high dimension space. In [11], the computation cost is made manageable by partitioning the data matrix into several smaller parts for biclustering, and the results are later merged to form the final biclusters. Later works proposed various strategies to address this shortcoming. For example, Zhao [10] used Hough transform to detect genes satisfying general linear model in pair of columns (column space) and then divide them into different patterns using additive and multiplicative pattern plot before combining them; GSGBC [12] first used Hough transform to find 2D linear coherent biclusters in column space and then applied graph spectrum analysis to obtain larger biclusters; Wang [13] used Hough transform to find 2D linear coherent patterns in column space and merge small biclusters into larger ones using hypergraph technique [13].

In gene expression data, the general linear model is biologically meaningful [2], [11]. Therefore, the number of biclustering algorithms based on the general linear model has increased significantly in recent years. In this study, we propose a two-phase algorithm that can find all linear coherent patterns. At the first phase, the proposed algorithm finds all linear relationships on a pair of columns. In the merging process, one creates larger sets of columns at the second phase. The paper is outlined as follow: the basic concepts and the proposed algorithm are described in Section 2; experiments and evaluations are given in Section 3 and finally, conclusions are drawn in Section 4.

2 Method

2.1 Hough Transform for Line Detection

The Hough transform (HT) is a method that detects lines and curves in images through a voting process in the parameter space [28], [15]. In the Cartesian coordinate system, an equation of a line is given by

$$y = kx + b \quad (1)$$

Given a set of point $\{(x_1, y_1), (x_2, y_2), \dots\}$, we would like to find a line (parameters k and b) that has the best fit to the given set of point. The basic idea of HT is dividing the range values of the parameters k and b into two dimensions grid (called accumulator). Therefore, each cell of accumulator corresponds with a pair values of (k, b) in the parameter space. For each point (x_i, y_i) , the accumulator cell it falls into is computed and voted on. Hence, the value of each accumulator cell gives the number of points

being on the corresponding line. Cells that received enough votes denote lines that are present in the image.

To avoid the problem of parameterization of vertical line using (1), the polar coordinate system is often used, i.e., the equation of a line in the polar coordinate system is defined as

$$r = x.\cos(\theta) + y.\sin(\theta) \tag{2}$$

where r is the distance from the origin to the line; θ is the angle between the line and the x -axis.

2.2 The Proposed Algorithm

If the Hough transform is used to detect a straight line and the number of accumulator cells (grid) in each dimension is A , the complexity of the Hough transform is A^2 . In general, the complexity of Hough transform for n dimensional hyperplane is A^n . So, the computational time of Hough transform for n dimension space is infeasible because the complexity is exponentially increasing. In order to overcome this problem, Zhao *et al.* [10] used Hough transform to find all straight lines in column-pair space ($n = 2$) and then applied a union-intersection operation to merge sub-biclusters into larger biclusters. The difference between our algorithm and [10] is in the second phase. In our algorithm, instead of a union-intersection operation, we find all possible higher dimension spaces through intersection operation.

The proposed algorithm takes as input an expression matrix, $\mathbf{A}(m \times n)$, where rows represent genes and columns describe conditions and output a set of biclusters that are linearly coherent. A bicluster is a sub-matrix $(G, C) \subseteq \mathbf{A}$, where G and C are sets of genes (rows) and conditions (columns), respectively. The proposed algorithm is a two-phase process. In the first phase, the Hough transform is used to find all rows which have linear relationships on pairs of columns in matrix \mathbf{A} . In the second phase, these pairs of columns are then merged to obtain larger set of columns on which the linear coherence occurs.

2.2.1 Phase 1

We apply Hough transform to find rows which have linear relationships in pairs of columns in matrix \mathbf{A} . Besides counting the number of votes for an accumulator cell, we also keep a record of the points (rows) that contribute to the vote of a cell. The result of this phase is a set of pairs of columns and list of points on which the linear relationships occur:

$$\{R^{1,2}, R^{1,3}, R^{1,4}, \dots, R^{2,3}, R^{2,4}, \dots, R^{i,j}, \dots, R^{n-1,n}\} \tag{3}$$

where $R^{i,j} = \{\text{points} \mid \text{their relationships are linear between columns } i \text{ and } j\}$.

The number of Hough transform required for this phase is $n(n-1)/2$. So, as the number of columns n in matrix \mathbf{A} increases, the complexity of this phase is $O(n^2)$. As the Hough transform on pairs of columns is independently performed, parallel computing can be applied to reduce the computational time at this phase.

2.2.2 Phase 2

The 2nd phase is a merging process based on the results that R^{ij} are set of points on the same lines. This process finds all points having linear relationships on at least three columns. The merging process is based on the two properties.

Property 1: If the two columns have linear relationship with another column at the same point, the relationship of the three columns is linear at this point.

Proof:

Given the linear relationship between columns c_i and c_j

$$c_i = k_{ij}c_j + b_{ij} \quad (4)$$

and between columns c_i and c_l

$$c_i = k_{il}c_l + b_{il} \quad (5)$$

we can easily infer the linear model between columns c_j and c_l as

$$c_j = \frac{k_{il}}{k_{ij}}c_l + \frac{b_{il} - b_{ij}}{k_{ij}} \quad (6)$$

Property 2: Given a set of columns having linear relationship at the same point, all nonempty subsets of this set must have the linear relationship.

Proof: we use induction proofs to prove.

From the 1st phase we obtain R^{ij} as the set of points on which two columns i and j are linearly related.

Let $n = 3$ (we prove that the 2nd property holds with a set of 3 columns)

If the column h has the linear relationship with both columns i and j at the same points, the following set must be nonempty ($R^{i,j,h} \neq \emptyset$):

$$R^{i,h} \cap R^{j,h} \cap R^{i,j} = R^{i,j,h} \quad (7)$$

Because $R^{i,j,h}$ is nonempty, all its subsets $R^{i,h}$, $R^{j,h}$ and $R^{i,j}$ are also nonempty.

Let $n = 4$: If the column l has the linear relationship with three columns $\{i, j, h\}$ at the same points, the following set must be nonempty ($R^{i,j,h,l} \neq \emptyset$):

$$R^{i,l} \cap R^{j,l} \cap R^{h,l} \cap R^{i,j,h} = R^{i,j,h,l} \quad (8)$$

We can easily observe that all subsets of $R^{i,j,h,l}$ are also nonempty.

$$\begin{aligned} R^{i,j,l} &= R^{i,j} \cap R^{i,l} \cap R^{j,l} \\ R^{j,h,l} &= R^{j,h} \cap R^{j,l} \cap R^{h,l} \\ R^{i,h,l} &= R^{i,h} \cap R^{i,l} \cap R^{h,l} \end{aligned} \quad (9)$$

and $R^{i,j,h}$

Assume the 2nd property hold as $n = k$, we prove that it is also satisfied as $n = k + 1$.

Assume the set of columns $\{i, j, h, l, \dots, k\}$ has the linear relationships at the same points. If the column $(k+1)$ has the linear relationship with a set of columns $\{i, j, h, l, \dots, k\}$ at the same points the following set must be nonempty ($R^{i,j,h,l,\dots,k,k+1} \neq \emptyset$):

$$R^{i,k+1} \cap \dots \cap R^{k,k+1} \cap R^{i,j,h,l,\dots,k} = R^{i,j,h,l,\dots,k,k+1} \tag{10}$$

and we easily obtain all subsets of $R^{i,j,h,l,\dots,k,k+1}$ are also nonempty. So, the 2nd property has just been proven.

Base on the two properties mentioned above, the 2nd phase can find all points having linear relationships on a set of columns. The pseudo-code of merging process is shown in Figure 1.

```

Input: A = {R1,2, R1,3, R1,4, ..., R2,3, R2,4, ..., Rij, ..., Rn-1,n}
Output: RS: sets of points having linear relationships on at least
three columns.

B = A;
RS = ∅;
While B ≠ ∅
Begin
    new_set = ∅;
    For i = 1 to |B|
    Begin
        For j = i+1 to |B|
        Begin
            c = bi ∩ bj;
            If c ≠ ∅ and c ∉ new_set then
            Begin
                Add c into new_set;
            End
        End
    End
    Add new_set into RS;
    B = new_set;
End
    
```

Fig. 1. Pseudo-code of merging process (|B| is cardinality of set B; b_i is an element of set B)

The 2nd phase is generally not a time-consuming task because it is based on intersection operation and linear coherent patterns are sparse.

3 Experiments

The two gene expression datasets, namely Yeast (Yeast *Saccharomyces cerevisiae* cell cycle) [16] and diffuse large-B-cell lymphoma [17] were used to evaluate the performance of the proposed algorithm and the following biclustering methods such as FABIA [9], ISA 2 [18], xMOTIF [19], Cheng–Church [20], Spectral biclustering [21], Plaid Model [22] were also compared. The Yeast dataset contains 2884 genes measured at 17 instances and the diffuse large-B-cell lymphoma was used to predict the survival after chemotherapy and contains 180 samples of 661 genes.

In Hough transform, each cell (r_i, θ_i) that received enough votes denote lines. In other words, if the number of points in the cell (r_i, θ_i) is greater than a predefined threshold, this is a line. In our experiments, this threshold was assigned to 12. Because the goal of the proposed algorithm is to find all linear biclusters, there are no limitations on bicluster size. These parameters were determined as optimal during processing of two gene expression datasets.

In order to assess the bicluster results of the two above biological data, we applied Gene Ontology (GO) [23] and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway [24]. The goal of the Gene Ontology Consortium is to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing and three independent ontologies are being constructed: biological process, molecular function and cellular component [23]. Among the existing tools for GO and KEGG pathway, we selected GO-TermFinder [25] and ClueGO [26].

All bicluster results found by the biclustering algorithms were enriched to three GO functional categories, namely biological process (GO BP), molecular function (GO MF), cellular component (GO CC), and KEGG pathway. For Yeast dataset, the biclusters obtained from the proposed algorithm and six existing biclustering methods that were enriched by three GO categories and KEGG pathway are shown in Tables 1–5. In this dataset, Spectral biclustering and Plaid Model did not produce any biclusters.

Table 1. Number of biclusters found by seven methods for yeast dataset

<i>Methods</i>	<i>Number of biclusters</i>
The proposed algorithm	6
xMOTIF	10
Spectral biclustering	0
Plaid Model	0
Cheng–Church	10
ISA 2	4
FABIA	10

Table 2. Number of biclusters enriched by GO BP

<i>Methods</i>	<i>p-value = 0.05</i>	<i>p-value = 0.01</i>
The proposed algorithm	6	6
xMOTIF	2	0
Spectral biclustering	0	0
Plaid Model	0	0
Cheng–Church	1	1
ISA 2	4	4
FABIA	7	6

Table 3. Number of biclusters enriched by GO MF

<i>Methods</i>	<i>p-value = 0.05</i>	<i>p-value = 0.01</i>
The proposed algorithm	6	6
xMOTIF	1	0
Spectral biclustering	0	0
Plaid Model	0	0
Cheng–Church	0	0
ISA 2	4	4
FABIA	7	6

Table 4. Number of biclusters enriched by GO CC

<i>Methods</i>	<i>p-value = 0.05</i>	<i>p-value = 0.01</i>
The proposed algorithm	6	6
xMOTIF	1	1
Spectral biclustering	0	0
Plaid Model	0	0
Cheng–Church	2	1
ISA 2	4	4
FABIA	6	6

Table 5. Number of biclusters enriched by KEGG pathway

<i>Methods</i>	<i>p-value = 0.05</i>	<i>p-value = 0.01</i>
The proposed algorithm	6	6
xMOTIF	2	2
Spectral biclustering	0	0
Plaid Model	0	0
Cheng–Church	0	0
ISA 2	4	4
FABIA	10	5

The biclusters of diffused large-B-cell lymphoma dataset given by several biclustering methods are listed in Figures 2–5 and Table 6. For this dataset, xMOTIF did not give any biclusters, and Cheng–Church considered whole database as a bicluster. While spectral clustering and Plaid Model formed only a single bicluster that was not significantly enriched by GO and KEGG pathway.

Table 6. Number of biclusters found by seven methods for lymphoma dataset

<i>Methods</i>	<i>Number of biclusters</i>
The proposed algorithm	16
xMOTIF	0
Spectral biclustering	1
Plaid Model	1
Cheng–Church	0
ISA 2	10
FABIA	5

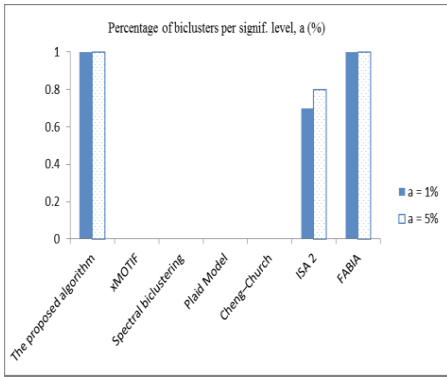


Fig. 2. Percentage of biclusters enriched by GO BP

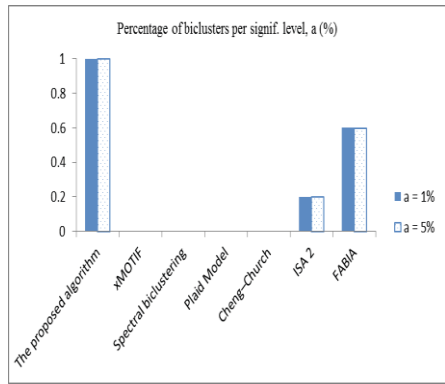


Fig. 3. Percentage of biclusters enriched by GO MF

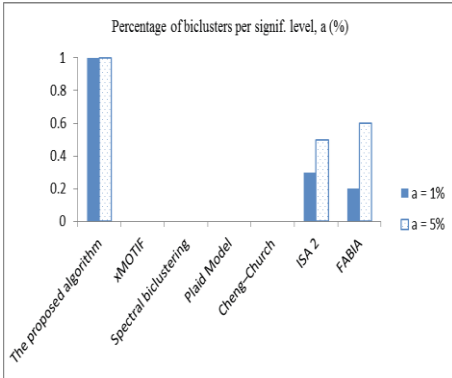


Fig. 4. Percentage of biclusters enriched by GO CC

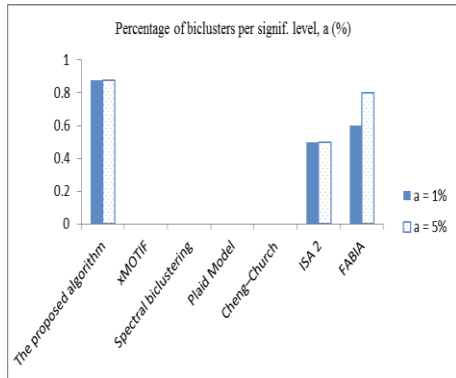


Fig. 5. Percentage of biclusters enriched by KEGG pathway

4 Conclusions

We have presented a geometric-based biclustering approach that is based on using the Hough transform. Our method finds all patterns, which have linear relationship on a pair of columns and then merge columns iteratively to obtain linear models on higher dimension spaces. The proposed algorithm was verified by two real gene expression datasets and was compared with several well-known biclustering methods. The results obtained from the proposed algorithm were found to be significantly enriched when evaluated using three GO categories and KEGG pathway.

References

1. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. *IEEE Transactions on Computational Biology and Bioinformatics* **1**, 24–45 (2004)
2. Zhao, H., Liew, A.W.C., Wang, D.Z., Yan, H.: Biclustering Analysis for Pattern Discovery: Current Techniques, Comparative Studies and Applications. *Current Bioinformatics* **7**(1), 43–55 (2012)
3. Li, G., Ma, Q., Tang, H., Paterson, A.H., Xu, Y.: QUBIC: a qualitative algorithm for analyses of gene expression data. *Nucleic Acids Research* **37**, e101 (2009)
4. Serin, A., Vingron, M.: DeBi: discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms for Molecular Biology* **6**, 18 (2011)
5. Huang, Q., Tao, D., Li, X., Liew, A.W.C.: Parallelized evolutionary learning for detection of biclusters in gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**, 560–570 (2012)
6. Huang, Q., Lu, M., Yan, H.: An evolutionary algorithm for discovering biclusters in gene expression data of breast cancer. In: *IEEE Congress on Evolutionary Computation*, pp. 829–834 (2008)
7. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge and Data Engineering* **18**, 590–602 (2006)
8. Chakraborty, A., Maka, H.: Biclustering of gene expression data using genetic algorithm. In: *Proceeding of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–8 (2005)
9. Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., Bijmens, L., Göhlmann, H.W., Shkedy, Z., Clevert, D.A.: FABIA: factor analysis for bicluster acquisition. *Bioinformatics* **26**, 1520–1527 (2010)
10. Zhao, H., Liew, A.W.C., Xie, X., Yan, H.: A new geometric biclustering algorithm based on the Hough transform for analysis of large-scale microarray data. *Journal of Theoretical Biology* **251**, 264–274 (2007)
11. Gan, X., Liew, A.W.C., Yan, H.: Discovering biclusters in gene expression data based on high-dimensional linear geometries. *BMC Bioinformatics* **9**, 209 (2008)
12. Wang, D.Z., Yan, H.: A graph spectrum based geometric biclustering algorithm. *Journal of Theoretical Biology* **317**, 200–211 (2013)
13. Wang, Z., Yu, C.W., Cheung, R.C.C., Yan, H.: Hypergraph based geometric biclustering algorithm. *Pattern Recognition Letters* **33**, 1656–1665 (2012)
14. Goldenshluger, A., Zeevi, A.: The Hough transform estimator. *Ann. Stat.* **32**, 1908–1932 (2004)
15. Illingworth, J., Kittler, J.: A survey of the Hough transform. *Comput. Vision Graphics Image Process* **44**, 87–116 (1988)
16. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. *Nat. Genet.* **22**, 281–285 (1999)
17. Rosenwald, A., et al.: The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New Engl. J. Med.* **346**, 1937–1947 (2002)
18. Ihmels, J., Bergmann, S., Barkai, N.: Defining transcription modules using large-scale gene expression data. *Bioinformatics* **20**, 1993–2003 (2004)
19. Murali, T.M., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. *Pacific Symposium on Biocomputing* **8**, 77–88 (2003)
20. Cheng, Y., Church, G.M.: Biclustering of expression data. In: *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103 (2000)

21. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research* **13**, 703–716 (2003)
22. Turner, H., Bailey, T., Krzanowski, W.: Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis* **48**, 235–254 (2005)
23. Ashburner, M., Ball, C.A., et al.: Gene Ontology: tool for the unification of biology. *Nature Genetics* **25**, 25–29 (2000)
24. Kanehisa, M., Goto, S.: KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28**, 27–30 (2000)
25. Boyle, E.I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J.M., Sherlock, G.: GO:TermFider—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* **20**, 3710–3715 (2004)
26. Bindea, G., Mlecnik, B., et al.: ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics* **25**, 1091–1093 (2009)
27. Gan, X., Liew, A.W.C., Yan, H.: Biclustering gene expression data based on a high dimensional geometric method. In: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (ICMLC 2005)*, vol. 6, pp. 3388–3393 (August 18 – 21, 2005)
28. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM* **15**, 11–15 (1972)

An Effective Biclustering Algorithm for Time-Series Gene Expression Data

Huixin Xu¹, Yun Xue^{2(✉)}, Zhihao Lu³, Xiaohui Hu², Hongya Zhao⁴,
Zhengling Liao², and Tiechen Li²

¹ School of Mathematical Sciences, South China Normal University, Guangdong, China
xuhuixin@genomics.cn

² School of Physics and Telecommunication, South China Normal University,
Guangdong, China
xueyun@scnu.edu.cn

³ Computer School, South China Normal University, Guangdong, China

⁴ Industrial Center, Shenzhen Polytechnic, Shenzhen, China
hy.zhao@szpt.edu.cn, {626332185, 1085206157}@qq.com

Abstract. The biclustering is a useful tool in analysis of massive gene expression data, which performs simultaneous clustering on rows and columns of the data matrix to find subsets of coherently expressed genes and conditions. Especially, in analysis of time-series gene expression data, it is meaningful to restrict biclusters to contiguous time points concerning coherent evolutions. In this paper, the BCCC-Bicluster is proposed as an extension of the CCC-Bicluster. An algorithm based on the frequent sequential mining is proposed to find all maximal BCCC-Biclusters. The newly defined Frequent-Infrequent Tree-Array (FITA) is constructed to speed up the traversal process, with useful strategies originating from Apriori Property to avoid redundant search. To make it more efficient, the bitwise operation XOR is applied to capture identical or opposite contiguous patterns between two rows. The algorithm is tested on the yeast microarray data. Experimental results show that the proposed algorithm is able to find all embedded BCCC-Biclusters, which are proven to reveal significant GO terms involved in biological processes.

Keywords: Biclustering · Time series · Gene expression data · Coherent evolution · Frequent sequential pattern mining · Bitwise operation

1 Introduction

Recently, numerous high-throughput developments of DNA chips generate a lot of massive gene expression data. Such data are represented as a matrix D of real-valued numbers (shown in Figure.1) with rows representing genes and columns representing different time points, different environmental conditions, different organs or different individuals. Each element represents the expression level of a gene under a specific condition (time).

$$\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{pmatrix}$$

Fig. 1. Gene expression data arranged in a matrix D

Traditional clustering techniques, such as K-means or hierarchical clustering, fail to detect local patterns in the data. Therefore, the biclustering, or subspace clustering, is proposed to overcome the aforementioned problem of traditional clustering methods. The biclustering performs simultaneous clustering on rows and columns of the data matrix to search subsets of genes and conditions which are coherently expressed. Clues are provided by subgroups which reveal some underlying biological processes. The biclustering is able to find local patterns in the form of subgroups of genes and conditions.

Cheng and Church [1] defines a bicluster as a subset of rows and a subset of columns (not necessarily contiguous) with a low mean squared residue H . They aims at finding large and maximal biclusters with H below a user-specified error threshold δ by using a greedy iterative search method. After the proposed of [1], different kinds of biclustering algorithms have been proposed [2,3,4,5,6,7,8,9]. Comprehensive surveys can be found in [10, 11].

In the analysis of time-series gene expression data, it is vital to find biclusters with coherent evolutions across contiguous columns. Time-series gene expression data focuses on the expression behavior of genes (rows) that have been exposed to a certain conditions at different instances of time (i.e. columns). There is an inherent sequential relationship between contiguous columns in these datasets. Each expression pattern shared by a group of genes in a contiguous subset of time points is a potentially relevant biological process [12]. Furthermore, it is often meaningful to determine coherent evolutions across columns of the data matrix without regarding to their exact values. It is because they usually span over a wide range when expression levels of genes are measured in different time points [13]. Due to noises and different expression levels of various genes, attentions have been paid on coherent evolutions across columns of the data matrix [2], [11], [14]. In a general model, it is assumed that all the genes in a given period are regulated either upwards or downwards. The biological meaning of continuous-column biclusters with coherent evolutions is presented in [12].

Consequentially, although major components of biclustering problems are NP-hard [15], these problems become tractable when expression levels are measured over time by restricting the analysis of biclusters to contiguous time points. There exist algorithms finding biclusters of continuous columns, for examples the CC-TSB (Column Coherent Time-Series Biclustering algorithm) [16] and the CCC-Biclustering (Continuous column coherent biclustering) [12], [17]. Our work is proposed based on the concept of CCC-Bicluster [12] and the maximal BCCC-Bicluster is proposed to extend the maximal CCC-Bicluster and discovers co-regulated genes with coherently bidirectional (similar or opposite) evolutions. The CCC-Biclustering aims at recognizing clusters of genes

whose expression levels rise and fall simultaneously throughout the given collection of continuous time points. However, it can not detect genes whose expression levels induce the reverse tendency over given time intervals. Such information is crucial because these genes may be influenced by the same regulators or involved in the same mechanics as those in CCC-Biclusters. The BCCC-Bicluster works as a generalization of the CCC-Bicluster to capture negative correlations among genes.

An algorithm based on the frequent sequential pattern mining is proposed to find all BCCC-Biclusters. The rest of the paper is organized as follows. The problem of BCCC-Bicluster mining is stated in Section 2. The proposed algorithm is described in Section 3. Experimental results are presented in Section 4. The paper is concluded in Section 5.

2 Problem Statement

Considering a gene expression dataset $D_{n \times m} = (R, C)$ with rows representing genes and columns representing different time points, while R and C denote the set of rows and columns in D respectively. Let d_{ij} denote the entry of D in row i and column j , representing the expression level of gene i at time j .

It is assumed that all the genes in a given bicluster are co-regulated (simultaneously upwards or downwards). In this experiment, we are interested only in the up-regulation and down-regulation of gene's activities over time. As the rows in a bicluster is supposed to be correlated or anti-correlated in a subset of continuous columns, the column difference between every two continuous columns is computed so as to identify this column subset. A simplified binary difference matrix $D'_{n \times (m-1)} = (R', C')$ is obtained as follows:

$$d'_{ij} = \begin{cases} 1, & d_{i, j+1} - d_{ij} \geq 0 \\ 0, & d_{i, j+1} - d_{ij} < 0 \end{cases} \tag{1}$$

After preprocessed, matrix D is transformed into a binary matrix D' and d'_{ij} represents the expression changes of gene i from time j to time $(j+1)$, where d'_{ij} equals 1 meaning up-regulation while d'_{ij} equals 0 meaning down-regulation.

In discovering co-regulated genes, CCC-Biclustering aims at identifying clusters of genes whose expression levels shows identical up-regulations or down-regulations over a period. However, it is also biological significance to know if there exists any gene i_2 whose expression levels are the mirror of gene i_1 's, as the activities of genes i_1 suppress that of genes i_2 , or vice versa[18]. It is obvious that two genes have reverse expression sequences are highly negative correlated. [19]. Hence we extend CCC-Bicluster to capture negative correlations among genes which is referred as BCCC-Bicluster.

Definition 1: A *bidirectional continuous column coherent bicluster (BCCC-Bicluster)*, $D'_{IJ} = (I, J)$, is a subset of rows $I = \{i_1, i_2, \dots, i_k\}$ ($I \subseteq R'$ and $k \leq n$) and a continuous subset of columns $J = \{r, r+1, \dots, s-1, s\}$ ($J \subseteq C'$ and $r < s \leq m-1$) from the binary matrix $D' = (R', C')$ such that either $d'_{ij} = d'_{lj}$ ($\forall i, l \in I$ and $j \in J$) or $d'_{ij} \neq d'_{lj}$ ($\forall i, l \in I$ and $j \in J$).

Definition 2: A *BCCC-Bicluster* is a maximal *BCCC-Bicluster* if it is not a proper sub-cluster of any *BCCC-Bicluster*.

A frequent continuous sequential pattern is common sequences whose support (giving the number of actual occurrences of identical or reverse sequences among given sequences) beyond a minimum support threshold N_{\min} . Such a frequent contiguous sequential pattern suggests the expression levels of co-regulated genes change (up-regulate or down-regulate) synchronously or oppositely over continuous time intervals. Based on such a perspective, the searching of maximal *BCCC-Biclusters* can be transformed into mining frequent sequential patterns, which is stated as follows:

Problem Statement (Maximal size-constrained *BCCC-Bicluster* mining problem): Given a data matrix D , a supporting row threshold N_{\min} , and a column threshold M_{\min} , find all maximal *BCCC-Biclusters* D'_{IJ} in D such that $|I| \geq N_{\min}$ and $|J| \geq M_{\min}$.

3 Algorithm

The algorithm is based on frequent sequential pattern mining. Given an n by m real-value data matrix D , an n by $(m-1)$ binary difference matrix D' is produced after preprocessed. For the n rows, there exist $(m-i+1) \times 2^i$ possible continuous sequential patterns of length i . Given a column threshold M_{\min} , there may be

$\sum_{i=M_{\min}}^{m-1} (m-i) \times 2^i$ continuous sequential patterns in n rows in the worst condition. It is

computationally complex and intractable to figure out the similar or opposite patterns of n rows at one time. To solve the problem, similar or reverse continuous patterns between two rows are pointed out at first and the rest rows are searched to check if such patterns are shared. In this way, the same or reverse patterns involving more than N_{\min} rows are found. To make it faster and easier to process, the bitwise operation XOR is performed to find out the similar or reverse patterns between two rows. The candidate sequences are recorded with corresponding indices of the first column, the last column and columns whose value in D' is 1, which not only keeps record of location and formation of candidates, but also saves storage space. Since the number of candidate sequences is still massive, a data structure named Frequent-Infrequent Tree-Array (FITA) is proposed to store the candidate sequences and fasten the traversal process. Consisting of Frequent Tree-Array (FTA) and Infrequent Tree-Array (ITA), Frequent-Infrequent Tree-Array (FITA) is applied with useful strategies based on Apriori properties to avoid redundant scanning. The flowchart of the algorithm is presented in Figure 2.

In this section, Bitwise XOR Operation is given to show how to find all continuous sequences with length larger than M_{\min} of every two rows in the subsection 3.1. And the Frequent-Infrequent Tree-Array (FITA) is described in detail in the subsection 3.2.

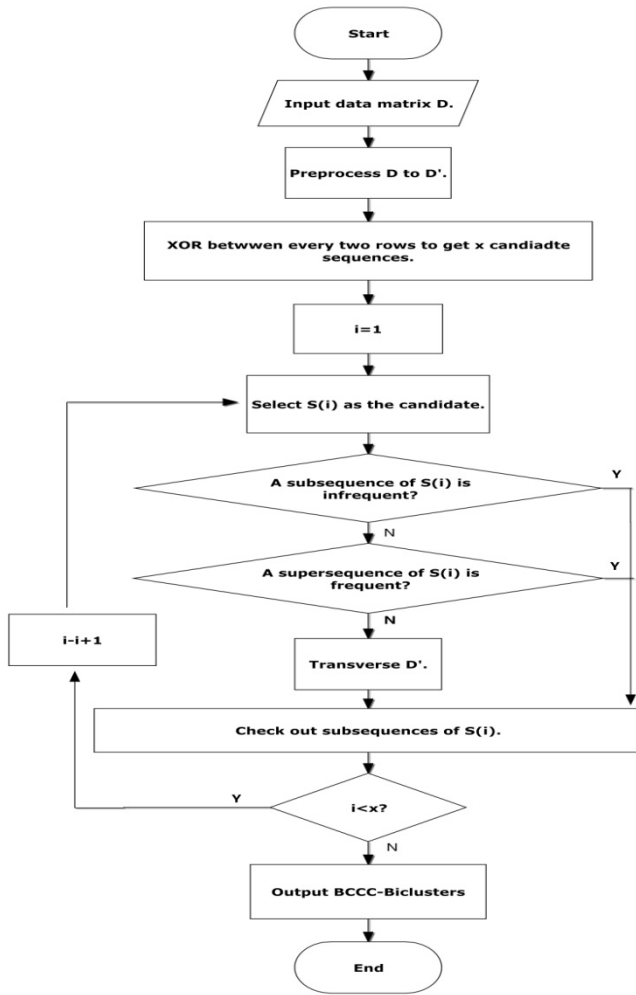


Fig. 2. The flowchart of the proposed algorithm

3.1 Bitwise XOR Operation

The use of bitwise operations is appropriate in order to facilitate the work and to improve the efficiency of the algorithm [20]. Applications of bitwise and its advantages are studied in [21,22,23]. There is no doubt that the bitwise operation XOR is much faster and easier to handle than the naive approach to compare two sequences because the bitwise XOR is a significantly lower cost operation than an integer comparison on most modern processors.

3.2 Frequent-Infrequent Tree-Array (FITA)

In this subsection, a new data structure is put forward to (1) organize the candidate sequences, (2) help to figure out supporting rows of frequent sequences, and (3) to avoid redundant search by using previous results. Certain strategies are proposed for faster traversal as well. The data structure is referred as Frequent-Infrequent Tree-Array (FITA), containing functionally independent but structurally similar Frequent Tree-Array (FTA) and Infrequent Tree-Array (ITA). Actually, FTA is an array whose elements consist of Frequent Trees (FT) storing all frequent continuous sequences. Similarly, ITA stores infrequent sequences in the form of Infrequent Tree (IT).

To construct the Frequent Tree-Array (FTA), an array H (short for “HEAD”) is built where $H[x] = x$ ($x \in N$, $1 \leq x \leq m-1$). Each element $H[x]$ in the array H contains a Frequent Tree (FT) storing the frequent contiguous sequences beginning with column x , thus x is known as “head”. Each frequent continuous pattern is represented as a path from root to leaf. The root is null and the first node, known as “tail”, shows the indices of last column of the contiguous sequence. Other nodes except leaf nodes of the path are represented by indices of column whose element equals 1 while the leaf stores row indices supporting its homogeneous or reverse continuous sequential pattern. What is more, there exists a pointer pointing to $H[x]$ when the tree of $H[x]$ is to be traversed. Infrequent Tree can be constructed in the similar way except that there is no need to report and memorize row indices in the leaf node.

4 Experimental Results

The algorithm is implemented by C language to find all Maximal size-constrained BCCC-Biclusters on the operating system Ubuntu 12.04 with Intel (R) Core (TM) i3 CPU and 4G memory. The algorithm is tested in yeast microarray data set. We evaluate the effectiveness of the algorithm with the p-value and size for the found biclusters.

4.1 Dataset

[Yeast Microarray Data] Yeast microarray data are a collection of 2884 genes under 17 contiguous time points. Each row corresponds to a gene and each column represents a time point under which the gene is developed. Each entry represents the relative abundance of the mRNA of a gene under a specific condition. All entries are integers lying in the range of 0–600. The missing values are replaced by random number between 0 and 800, as in [24]. The data are available at <http://arep.med.harvard.edu/biclustering/>.

4.2 Results From the Yeast Microarray Data

Parameters N_{\min} and M_{\min} are set to be 2 and 2 respectively in this experiment. In general, we find 16714 BCCC-Biclusters in less than 12 seconds. There are 16628 biclusters whose number of continuous columns is no less than 4, of which 33.93%, 5642 BCCC-Biclusters, contain similar and opposite evolution (up-regulation or

down-regulation). The biggest one in rows is a bicluster with 702 rows and 5 contiguous columns.

Here we present the 1054th BCCC-Bicluster BC consisting of similar and reverse expression patterns, as shown in Figure.3. The 1054th bicluster consists of 628 genes and 5 continuous columns (Column 10 to Column 14). Figure.3 (a) shows the bicluster plotted with x-axis representing the columns, y-axis representing the expression values and each line representing a gene. Figures 3(b) and 3(c) show the 1054th BCCC-Bicluster. BC is made up of 2 CCC-Biclusters: BC1 and BC2. BC1 consists of 525 genes whose pattern is 1010, while BC2 involves 103 genes with the pattern 0101. Such biclusters would be missed by CCC-Biclustering because CCC-Biclustering in [11] does not include coherently opposite evolution.

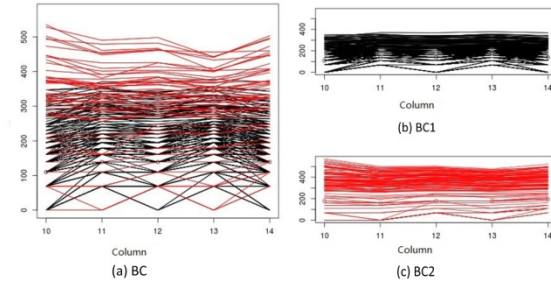


Fig. 3. The 1054th BCCC-Bicluster BC consisting of two CCC-Biclusters, BC1 and BC2

Our algorithm is able to detect overlapping BCCC-Biclusters as well. An example is presented in Figure.4. The 6800th BCCC-Bicluster A (the upper one in the red-lined area) is a 694 by 5 matrix consisting of a 236 by 5 matrix A_1 whose sequential pattern is 0101 and a 458 by 5 matrix A_2 whose sequential pattern is 1010. The 4572nd BCCC-Bicluster B (the lower one in the blue-lined area) is a 640 by 4 matrix consisting of a 78 by 4 matrix B_1 whose sequential pattern is 0101 and a 78 by 5 matrix B_2 whose sequential pattern is 1010. $A_1 \cap B_1$ consists of a 47 by 4 matrix whose pattern is 0101 while $A_2 \cap B_2$ consists of a 246 by 4 matrix whose pattern is 1010. So the overlapping part of A and B is a 293 by 4 matrix.

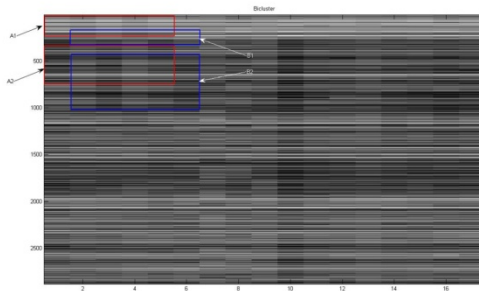


Fig. 4. The overlapping of the 6800th and 4572nd BCCC-Bicluster

The CCC-Biclustering [11] fails to report expression patterns revealing negative correlations. But such patterns may be biologically meaningful. To study the biological relevance of BCCC-Biclusters, we make use of the gene ontology (GO) annotations. GOToolBox [25] on <http://genome.crg.es/GOToolBox/> is used for annotation analysis, with which the gene sets in the bicluster can be studied through different GO categories. Some BCCC-Biclusters including negative correlations are extracted for analysis. The biclusters chosen vary in sizes and patterns. The summary is provided in TABLE 1. The information of found biclusters is given in Column 1 to 4, presenting their IDs, corresponding patterns, and numbers of genes and time points. With GO terms enriched, the number of GO terms is counted and recorded in Column 5 and 6 when its corresponding p-value is lower than 0.01 or between 0.01 and 0.05. The best p-value, referred to the lowest one, is reported in Column 7. It can be seen that most biclusters selected, even the ones with small sizes (ID 9327), relate to more than ten GO terms whose p-value is highly statistically significant ($p < 0.01$), which shows that BCCC-Biclusters found may be biologically meaningful.

Table 1. The Summary of go terms of extracted Bccc-Biclusters

ID	Expression patterns	GO terms summary				
		#Genes	# Time points (first-last)	C5	C6	Best p-value (10^{-2})
849	0101+1010	620+82	5(7-11)	11	9	0.320
255	0101+1010	595+99	5(9-13)	2	6	0.106
1729	1101+0010	40+617	5(6-10)	17	0	1.559
6800	0101+1010	236+457	5(3-7)	11	2	0.441
4307	1111+0000	80+106	5(13-17)	8	11	0.077
3826	00101+11010	50+17	6(9-14)	11	8	0.243
9327	110110101010 +001001010101	3+1	13(4-16)	4	8	0.582

C5: #p-values (<0.01), showing the number of GO terms with $p < 0.01$.

C6: #p-values (0.01, 0.05), showing the number of GO terms with $0.01 < p < 0.05$.

5 Conclusions

In this paper, an algorithm is implemented to find co-regulated genes in time-series gene expression data. The BCCC-Bicluster is proposed as an extension of the CCC-Bicluster to detect genes which perform bidirectional evolutions. The BCCC-Biclustering problem is transformed into a frequent sequential pattern mining problem. We apply the bitwise XOR operation and the new data structure: Frequent-Infrequent Tree-Array (FITA) to speed up the mining process. The biological meanings of BCCC-Biclusters are presented with statistically enriched GO terms.

Acknowledgement. The authors thank gratefully for the colleagues participated in this work and provided technical supports. This work is supported by Guangdong Science and Technology Department under Grant No.2009B090300336, No.2012B091100349; Guangdong Economy & Trade Committee under Grant No. GDEID2010IS034; Guangzhou Yuexiu District science and Technology Bureau under Grant No 2012-GX-004; National Natural Science Foundation of China (Grant No:71102146, No.3100958); Science and Technology Bureau of Guangzhou under Grant No. 2011J4300046; Research supported in part by the Guangdong Nature Science Fund (No. S2012010010661).

References

1. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 93–103. AAAI Press (2000)
2. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: RECOMB 2002: Proceedings of the Sixth Annual International Conference on Computational Biology, pp. 49–57. ACM, New York (2002)
3. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18**, S136–S144 (2002)
4. Divina, F., Aguilar-Ruiz, J.S.: A multi-objective approach to discover biclusters in microarray data. In: GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 385–392. ACM, New York (2007)
5. Gu, J., Liu, J.S.: Bayesian biclustering of gene expression data. *BMC Genomics* **9**(suppl. 1), S4 (2008)
6. Lazzeroni, L., Owen, A.: Plaid models for gene expression data. *J Statistica Sinica* **12**, 61–86 (2002)
7. Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., Zitzler, E.: Bicat: a biclustering analysis toolbox. *Bioinformatics* **22**(10), 1282–1283 (2006)
8. Bleuler, S., Prelic, A., Zitzler, E.: An ea framework for biclustering of gene expression data. In: Proceedings of Congress on Evolutionary Computation, pp. 166–173 (2004)
9. Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression data. In: BIBE 2003: Proceedings of the 3rd IEEE Symposium on Bioinformatics and Bio Engineering, pp. 321. IEEE Computer Society, Washington, DC (2003)
10. Prelic, A., Bleuler, S., Zimmermann, P., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* **22**(9), 1122–1129 (2006)
11. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **1**(1), 24–45 (2004)
12. Madeira, S.C., Oliveira, A.L.: A Linear Time Biclustering Algorithm for Time Series Gene Expression Data. In: Casadio, R., Myers, G. (eds.) WABI 2005. LNCS (LNBI), vol. 3692, pp. 39–52. Springer, Heidelberg (2005)
13. Murali, T.M., Kasif, S.: Extracting Conserved Gene Expression Motifs from Gene Expression Data. In: Proc. Pacific Symp. Biocomputing, vol. 8, pp. 77–88 (2003)

14. Liu, J., Yang, J., Wang, W.: Biclustering in gene expression data by tendency. In: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference, CSB 2004, August 16-19, pp. 182–193 (2004)
15. Peeters, R.: The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics* **131**(3), 651–654 (2003)
16. Zhang, Y., Zha, H., Chu, C.-H.: A time-series biclustering algorithm for revealing co-regulated genes. In: International Conference on Information Technology: Coding and Computing, ITCC 2005, April 4-6, vol. 1, pp. 32–37 (2005)
17. Madeira, S.C., Teixeira, M.C., Sá-Correia, I., Oliveira, A.L.: Identification of Regulatory Modules in Time Series Gene Expression Data using a Linear Time Biclustering Algorithm. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (March 21, 2008)
18. Cheung, L., Cheung, D.W., Kao, B.: On mining micro-array data by Order-Preserving Submatrix. *International Journal of Bioinformatics Research and Applications* **3**, 42–64 (2007)
19. Gao, B.J., Griffith, O.L., Ester, M., Xiong, H., Zhao, Q., Jones, S.J.M.: On the Deep Order-Preserving Submatrix Problem: A Best Effort Approach. *IEEE Trans. Knowl. Data Eng.* **24**, 309–325 (2012)
20. Yordzhev, K.: An Example for the Use of Bitwise Operations in programming. *Mathematics and Education in Mathematics* **38**, 196–202 (2009)
21. Gottesman, D.: A theory of fault-tolerant quantum computation. *Phys. Rev. A* **57**, 127–137 (1998)
22. Hall, K.L., Rauschenbach, K.A.: 100-Gbit/s bitwise logic. *Opt. Lett.* **23**(16), 1271–1273 (1998)
23. Tan, K.-L., Eng, P.-K., Ooi, B.C.: Efficient progressive skyline computation. In: Proc. of the Conf. on Very Large Data Bases, Rome, Italy (September 2001)
24. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 93–103. AAAI Press (2000)
25. Martin, D., Brun, C., Remy, E., Mouren, P., Thieffry, D., Jacq, B.: GOToolBox: functional investigation of gene datasets based on Gene Ontology. *Genome Biology* **5** (12R101) (2004). <http://burgundy.cmmt.ubc.ca/GOToolBox/>

Multiple Orthogonal K-means Hashing

Ziqian Zeng^(✉), Yueming Lv, and Wing W.Y. Ng

School of Computer Science and Engineering,
South China University of Technology, Guangzhou 510006, China
celine.ziqianzeng@gmail.com, wingng@ieee.org

Abstract. Hashing methods are efficient in dealing with large scale image retrieval problems. Current hashing methods, such as the orthogonal k-means, using coordinate descent algorithm to minimize quantization error usually yield unstable performance. It is because the coordinate descent algorithm only provides a local optimum solution. The orthogonal k-means develops a new model with a compositional parameterization of cluster centers to efficiently represent multiple centers. The objective of the orthogonal k-means is to minimize the quantization error by using the coordinate descent algorithm to find the optimal rotation, scaling and translation on descriptor vectors of images. The performance of the orthogonal k-means is dependent on the initialization of the rotation matrix. In this work, we propose the multiple ok-means hashing method to reduce the instability of performance of the orthogonal k-means hashing. For large scale retrieval problems, standard multiple hash tables methods using M tables require M times storage in comparison to single hash table schemes. We propose a binary code selection scheme to reduce the storage of the multiple orthogonal k-means to use the same size of storage as for single table's. Experimental results show that the proposed method outperforms ok-mean using the same size of storage.

Keywords: Approximate nearest neighbor search · Hashing · Multiple hash tables · Large scale image retrieval

1 Introduction

The Internet offers a huge number of image resources. Given a query image interested by a user, it is time consuming to find the most similar images from such a huge database consisting of millions of images. Fortunately, usually it is not necessary to find the most similar image. A "good approximation" of the nearest neighboring image can usually be acceptable by human users, which is the key idea of approximate nearest neighbor search (ANN) [1]. Recently, hashing based approximate nearest search has become popular because of its sublinear search time. Hashing methods aim to map the real-valued feature vector describing the image to a binary code by using hashing functions [2]. Images are stored in binary codes in the memory for searching to minimize time consuming hard disk access. When given a query image, it is first encoded to a binary code, and then the Hamming distances between its binary code and the binary codes of images in the database are computed. Finally, images are returned to users

ranked by the ascending order of their Hamming distances. The computation of Hamming distance is fast and easy because it only involves XOR and bit operations [2]. Even performing a linear search in the database, hashing based ANN search is still very fast [2]. By database indexing and other techniques, no linear search is needed and the group of images with small Hamming distances from the query image could be returned in a sublinear time.

Recently, many hashing methods have been proposed to find similarity preserving binary codes. The Locality Sensitive Hashing (LSH) [4, 5, 6] uses random projections to obtain binary codes and guarantees that similar images are mapped to the same hash bucket with a high probability. The probability is directly proportional to their similarity. Many variants of the LSH have been proposed, for example the Locality-Sensitive binary codes from Shift-Invariant Kernels (SKLSH) [7] which uses random Fourier features to preserve Euclidean distance between two images. Both the LSH and the SKLSH need long hash code to preserve a good similarity; hence both of them suffer from severe redundancy. So, some hashing methods are proposed to find compact similarity-preserving binary codes [3, 8 - 13]. The ITQ [8, 9] aims to find a rotation of zero-centered data (i.e. images) to minimize the quantization error of mapping this data to the vertices of a zero centered binary hypercube, and proposes a simple and efficient alternating minimization algorithm to accomplish this task. Initializing with a random rotation matrix, the ITQ [8, 9] adopts a k-means like procedure to find the local minimum of the quantization loss. In addition, there are some other methods related to the k-means clustering [14], e.g. the Cartesian k-means [15]. Authors of [15] proposed two types of k-means based models: the Orthogonal K-means (ok-means) and the Cartesian k-means (ck-means). The ck-means does not generate binary hash codes and does not compare similarity based on Hamming distance. This is because the binary code generated by the ck-means does not preserve similarity in Euclidean distance and therefore cannot be used to compare the similarity between images using Hamming distance. In contrast, the ok-means obtains the similarity preserving binary code like other hashing methods, so we focus on ok-means in this paper. The objective of ok-means is to find a rotation, a scaling and a translation to minimize the quantization error by using coordinate descent algorithm. The performance of both the ITQ and the ok-means depend on the random initialization of the rotation matrix R . So, we propose the Multiple Orthogonal K-means hashing method (mok-means) to reduce this dependence and further enhance the image retrieval performance.

Compared with aforementioned hashing methods using a single hash table, multiple hash table based methods generally yield a better performance. The LSH [4-6] can be easily extended to multiple hashing by randomly and independently creating multiple hash tables to enlarge the probability of mapping similar images to similar hash codes. However, a large number of hash tables is needed to achieve a satisfactory performance. In contrast to constructing multiple hash tables independently, the Complementary Hashing (CH) [16] adopts a boosting based learning framework to construct multiple hash tables with similarity information of training images.

The major disadvantage of multiple hash tables is the increased storage cost which could be prohibitive for large scale image retrieval tasks. However the proposed

mok-means with multiple hash tables avoids a large storage cost. Actually, the storage cost of the mok-means is the same as methods with a single hash table only.

We introduce the ok-means in details and propose the mok-mean in Sections 2 and 3, respectively. Experimental results on two real databases will be shown and discussed in Section 4. We conclude this work in section 5.

2 The Orthogonal K-means

The Orthogonal k-means can be viewed as an extension of k-means clustering for constructing clusters.

Given a dataset $S \equiv \{x_j\}_{j=1}^n$, where each x_j is a d -dimensional real vector, the k-means algorithm partitions n data points into k clusters. For each cluster, there is a center point serving as its representative. Let $C \in \mathfrak{R}^{d \times k}$ be a matrix whose columns are k cluster centers, i.e., $C = [c_1, c_2, \dots, c_k]$. The k-means minimizes the within cluster squared distances as follows:

$$\ell_{\text{k-means}}(C) = \sum_{x \in S} \min_i \|x - c_i\|_2^2 \quad (1)$$

$$= \sum_{x \in S} \min_{b \in \mathbf{H}_{1/k}} \|x - Cb\|_2^2 \quad (2)$$

where $\mathbf{H}_{1/k} \equiv \{b \mid b \in \{0,1\}^k \text{ and } \|b\| = 1\}$.

As an extension of the k-means clustering, the ok-means [15] partitions n points into 2^m clusters where clusters are represented as an additive combination of the columns of C . So, the ok-means minimizes the within cluster squared distance as follows:

$$\ell(C) = \sum_{x \in S} \min_{b \in \mathbf{H}_m} \|x - Cb\|_2^2 \quad (3)$$

where $C \in \mathfrak{R}^{d \times m}$ where $C^T C$ is a diagonal matrix, and $b \in \mathbf{H}_m \equiv \{0,1\}^m$.

To further reduce the quantization error, authors of [15] introduce three transformation parameters, namely, rotation R , scaling D , and translation μ . Finally the objective function of the ok-means is as follows:

$$\ell_{\text{ok-means}}(C, \mu) = \sum_{x \in S} \min_{b' \in \mathbf{H}_m'} \|x - \mu - Cb'\|_2^2 \quad (4)$$

where $C \equiv RD$, $b' \in \mathbf{H}_m' \equiv \{-1,+1\}^m$, $R \in \mathfrak{R}^{d \times m}$ where $R^T R = I_m$, and $D \in \mathfrak{R}^{m \times m}$ is a diagonal and positive definite matrix.

In order to learn transformation parameters: R , D , and μ , The ok-means re-writes the objective function in a matrix form:

$$\ell_{\text{ok-means}}(B', R, D, \mu) = \|X - \mu \mathbf{1}^T - RDB'\|_f^2 \quad (5)$$

$$= \|X' - RDB'\|_f^2 \quad (6)$$

where $X \in \mathfrak{R}^{d \times n}$, $X' \equiv X - \mu \mathbf{1}^T$, $B' \in \{-1, +1\}^{m \times n}$, and $\|\cdot\|_f$ denotes the Frobenius norm.

Firstly, R and μ are randomly initialized. Then, the coordinate descent algorithm is used to minimize Equation (6) iteratively using the following three steps [15]:

- Fix R and μ , then optimize B' and D : from Equation (6), one can easily obtain Equation (7):

$$\ell_{\text{ok-means}} = \|R^T X' - DB'\|_f^2 + \|R^{\perp T} X'\|_f^2 \quad (7)$$

where columns of R^{\perp} span the orthogonal complement of the column space of R .

The second term is fixed, so we just need to focus on the first term. To minimize the first term, B' should be $+1$ whenever $R^T X' \geq 0$ and -1 otherwise. For D , the optimal d_i (where $d_i = D_{ii}$) is determined by the mean absolute value of the elements in the i^{th} row of $R^T X'$:

$$B' \leftarrow \text{sgn}(R^T X') \quad (8)$$

$$D \leftarrow \text{Diag}(\underset{\text{row}}{\text{mean}}(\text{abs}(R^T X'))) \quad (9)$$

- Fix B' , D and μ , then optimize R : find R that minimizes $\|X' - RDB'\|_f^2$, subject to $R^T R = \mathbf{I}_m$. This is equivalent to an Orthogonal Procrustes problem [17], and can be solved using a singular value decomposition. R is not a square matrix and authors of [15] point out that by adding $d - m$ rows of zeros to the bottom of D makes DB becoming a $d \times n$ matrix. Then R is a square and orthogonal matrix, and can be estimated using a singular value decomposition.

- Fix B' , D , R , optimize μ , the optimal μ is determined by the column average of $X - RDB'$:

$$\mu \leftarrow \underset{\text{column}}{\text{mean}}(X - RDB') \quad (10)$$

It is shown that there are two ways to estimate the distance between two vectors, \mathbf{u} and \mathbf{v} [15, 18]. The Symmetric Quantizer Distance (SQD) approximates the distance between the two vectors by $\|q(\mathbf{v}) - q(\mathbf{u})\|$ while the Asymmetric Quantizer Distance (AQD) approximates it by $\|\mathbf{v} - q(\mathbf{u})\|$ where $q(\cdot)$ is generic quantizer. In this paper, we are only interested in the SQD. In the ok-means model, the quantization of an input x is:

$$q_{\text{ok}}(x) = \mu + RD \text{sgn}(R^T(x - \mu)) \quad (11)$$

The index of x is:

$$b' = \text{sgn}(R^T(x - \mu)) \quad (12)$$

Given two indices, namely, $b_1', b_2' \in \{-1, +1\}^m$, the symmetric ok-means quantizer distance is:

$$SQD_{ok}(b_1', b_2') = \|\mu - RDb_1' - \mu - RDb_2'\|_2^2 \quad (13)$$

$$= \|D(b_1' - b_2')\|_2^2 \quad (14)$$

Actually, the SQD_{ok} is a weighted Hamming distance, and the experiments in [15] show that the Hamming distance ($\|b_1' - b_2'\|_2^2$) and the weighted Hamming distance yield similar results. So in this paper, we just adopt $\|b_1' - b_2'\|_2^2$ rather than the SQD_{ok} . Hence, the ok-means can be applied to hashing based ANN search. Given a query image y , a binary code b_y' is generated, then the Hamming distances between b_y' and indices (columns of B') of images stored in the database are computed to find the ANNs of the query image.

3 Multiple Orthogonal k-Means Hashing

The proposed Multiple Orthogonal K-means Hashing method (mok-means) is proposed based on the observation that in a large dataset, quantization errors of some data points are small under a set of transformation parameters P_1 , while quantization errors of other data points are small under another set of transformation parameters P_2 . Figure.1 provides a graphical illustration of this observation.

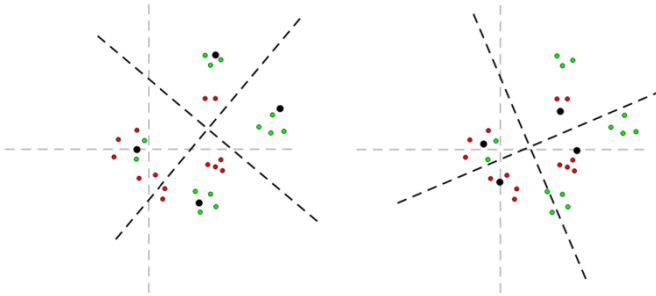


Fig. 1. The two sub-figures (left and right) show that the 2D data points (red and green small dots) are quantized with parameters P_1 and P_2 , respectively. Cluster centers are depicted by big black dots, and clusters boundaries are depicted by black dashed lines. The green dots are well quantized (quantization error is small) with parameters P_1 while red dots are well quantized with parameters P_2 .

The key idea of the mok-means hashing is to use T sets of parameters rather than using a single set of parameters as ok-means does to maintain a low overall quantization error.

The objective function of the mok-means is:

$$\ell_{\text{mok-means}}(X_i) = \sum_{i=1}^T \|X_i - \mu_i \mathbf{1}^T - R_i D_i B_i'\|_f^2 \quad (15)$$

where X_1, X_2, \dots, X_T are disjoint subsets of $S \equiv \{x_j\}_{j=1}^n$, and R_i, D_i, μ_i are a set of parameters generated by the ok-means algorithm. B_i' can be obtained by X_i, R_i, μ_i , i.e., $B_i' = \text{sgn}(R_i^T (X_i - \mu_i))$.

The learning algorithm of mok-means can be divided into two steps:

- Run the ok-means algorithm T times. Every time, the ok-means will generate three parameters, namely R, D, μ . Let $\mathbf{P} = \{P_i\}_{i=1}^T$, where $P_i \equiv \{R_i, D_i, \mu_i\}$, and R_i, D_i, μ_i are the output of i^{th} ok-means run.

- Determine subsets X_i for $i=1, \dots, T$. If an image (x) yields the smallest quantization error using parameter P_i (i.e., $\|x - \mu_i - R_i D_i \text{sgn}(R_i^T (x - \mu_i))\|_2^2$), then it belongs to the X_i , i.e. $x \in X_i$. For every data point x , find table index t as follows:

$$t = \arg \min_i \|x - \mu_i - R_i D_i \text{sgn}(R_i^T (x - \mu_i))\|_2^2, \text{ subject to } i=1, \dots, T \quad (16)$$

The mok-means constructs T hash tables and B_i' denotes the binary codes of the i^{th} hash table. Given that X_1, X_2, \dots, X_T are disjoint, so the total storage cost of the mok-means hashing is reduced to be the same as the storage cost of the ok-means.

Given a query y , all T hash codes of y are generated: $b_{y1}', b_{y2}', \dots, b_{yT}'$ where $b_{yi}' = \text{sgn}(R_i^T (y - \mu_i))$. Then, Hamming distances between b_{yi}' and B_i' stored in i^{th} hash table are computed, separately. Finally, ANNs of the query from each X_i are found according to the Hamming distance in ascending order.

In comparison to the ok-means, the mok-means generate T binary codes for the query image. The time of generating T binary codes is fast and negligible. On the other hand, in comparison to other multiple hashing based methods which requires T times of Hamming distance computation and T times of storage. The mok-means only compare binary codes of images in the database once and stores one binary code per image. For each image, only the binary code yielding the smallest quantization error among T tables is stored. Overall, the mok-means provides a better retrieval results while does not require extra storage.

4 Experimental Results

In experiments, the mok-means is compared with the ok-means [15], the ITQ [8,9], the ITQ-RFF [9], the MLSH[4-6] (LSH with multiple hash tables) and the LSH[4-6] on the SIFT1M database with 128-dimensional SIFT descriptors as in [15] and the CIFAR10 database with 320-dimensional GIST feature set as in [8-9]. The ITQ-RFF maps the data onto a higher dimensional space using random Fourier features to capture nonlinear structure in the data. The MLSH perform multiple LSH random projections to obtain multiple hash tables. The SIFT1M database [19] has about 1 million of images. The CIFAR10 database [20] is a subset of the Tiny Images Database and consists of 60,000 images. In our experiments of both databases, we randomly selected 1000 images as the test query set and the others serve as training set. Queries are made to retrieve images from the training image set.

4.1 Initialization

Although authors of the ok-means [15] claim that no pre-processing is needed for dimensionality reduction, the codes provided by authors provide both versions of with and without PCA (Principal Component Analysis) dimensionality reduction. The version with PCA dimensionality reduction uses the PCA dimensionality reduction as a pre-processing to speed-up the whole process. We find that the version with PCA outperforms the one without PCA with the same number of iterations. Hence, in our experiments, we adopt the version with PCA to implement both the ok-means and the mok-means. For the ok-means and the mok-means hashing, initial R is a random rotation of the first m principal directions of the training data, and initial μ is the mean value of the data as in [15]. For both the ITQ and the ITQ-RFF, data points are pre-processed to have zero means as required in [8, 9]. In addition, for both the MLSH and the LSH, data points are also normalized to have zero means.

4.2 Parameters

For all the experiments, we use the average value of Euclidean distances of the 50th nearest neighbor as the threshold of the ground truth which is the same as in [8-9]. For the ITQ and the ok-means, we use the codes provided by authors, and the number of iterations is set to 50 (default value in [8, 9]). The number of iterations of the mok-means is also set to 50 for fair comparisons. We implement programs for the ITQ-RFF, the MLSH and the LSH according to the corresponding papers. For the mok-means and the MLSH, the number of hash tables T is set to be 15.

4.3 Results

Figures 2 and 3 show recall-precision curves of different hashing methods for the SIFT1M and the CIFAR10 databases, respectively, using 16-bit, 24-bit and 32-bit hash codes. The results consistently favor the mok-means on both databases. The

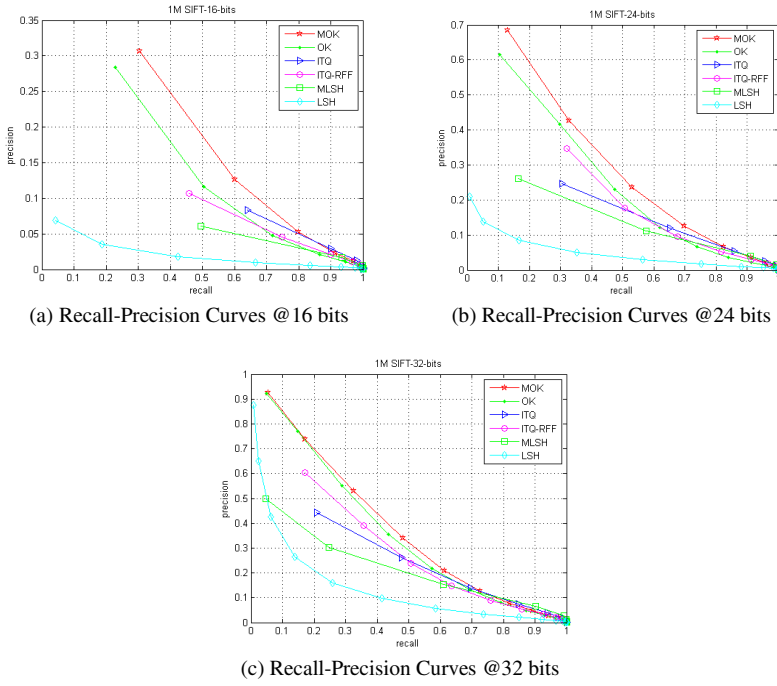


Fig. 2. Comparisons on the SIFT1M Database

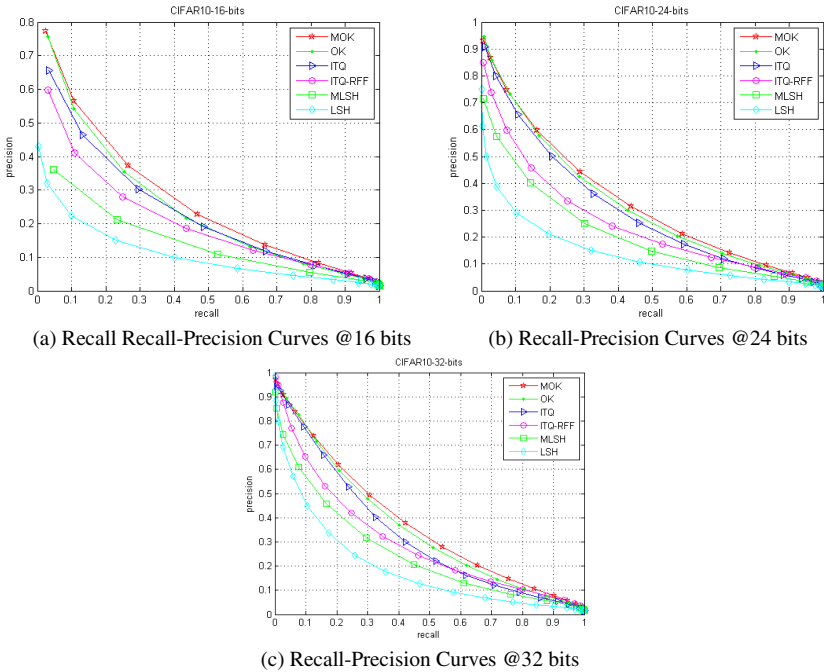


Fig. 3. Comparisons on the CIFAR10 database

ok-means performs better than the ITQ, which is consistent with the results shown in [15]. The ITQ outperforms the ITQ-RFF with 16, 24, 32 hash bits for experiments using the CIFAR10 database, which is also consistent with the results in [9]. The LSH yields a poor performance with short code length and the MLSH yields a better performance than the LSH by constructing multiple hash tables.

5 Conclusion

The multiple orthogonal k-means (mok-means) hashing aims to lower the quantization error as much as possible by constructing multiple hash tables. For large scale image retrieval tasks, using multiple hash tables is very expensive in terms of storage, but we propose a novel multiple hash tables scheme which yields a storage cost to be the same as a single hash table hashing. We divide the dataset into T disjoint sub-sets according to hash table yielding the smallest quantization error of an image. In this way, the mok-means share the benefit of both high recall of multiple hashing and small storage cost of single hashing.

Acknowledgements. This work is supported by a National Natural Science Foundation of China (61272201), a Fundamental Research Funds for the Central Universities (10561201465) and a Student Research Project of South China University of Technology (105612014S467).

References

1. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching fixed dimensional. *Journal of the ACM* **45**(6), 891–923 (1998)
2. Wu, C., Zhu, J., Cai, D., Chen, C., Bu, J.: Semi-supervised nonlinear hashing using bootstrap sequential projection learning. *IEEE Transactions on Knowledge and Data Engineering* **25**(6), 1380–1393 (2013)
3. Gordo, A., Perronnin, F.: Asymmetric distances for binary embeddings. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 729–736 (2011)
4. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p -stable distributions. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry, pp. 253–262. ACM (2004)
5. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, pp. 518–529 (1999)
6. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2006, pp. 459–468 (2006)
7. Raginsky, M., Lazechnik, S.: Locality-Sensitive Binary codes from Shift-Invariant Kernels. In: Advances in Neural Information Processing Systems (NIPS), 22, pp. 2130–2137 (2009)
8. Gong, Y., Lazechnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 817–824 (2011)

9. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(12), 2916–2929 (2013)
10. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: *Advances in Neural Information Processing Systems (NIPS)* **9**, pp. 1753–1760 (2008)
11. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: *Advances in Neural Information Processing Systems (NIPS)* **22**, pp. 1042–1050 (2009)
12. Norouzi, M., Blei, D.M.: Minimal loss hashing for compact binary codes. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-2011)*, pp. 353–360 (2011)
13. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2074–2081 (2012)
14. Lloyd, Stuart P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory* **28**(2), 129–136 (1982)
15. Norouzi, M., Fleet, D.J.: Cartesian k-means. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3017–3024 (2013)
16. Xu, H., Wang, J., Li, Z., Zeng, G., Li, S., Yu, N.: Complementary hashing for approximate nearest neighbor search. In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 1631–1638 (2011)
17. Schönemann, P.: A generalized solution of the orthogonal procrustes problem. *Psychometrika* **31**(1) (1966)
18. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(1) (2011)
19. <http://corpus-texmex.irisa.fr/>
20. Krizhevsky, A.: Learning multiple layers of features from tiny images. MSc Thesis, Univ. Toronto (2009)

Application to Recognition

Recognizing Bangladeshi Currency for Visually Impaired

Mohammad M. Rahman¹, Bruce Poon^{1,2(✉)}, M. Ashraful Amin^{1,3}, and Hong Yan^{2,3}

¹ Computer Vision and Cybernetics Research, Computer Science and Engineering,
Independent University Bangladesh (IUB), Dhaka, Bangladesh
{motiur.rahman0, aminmdashraful}@gmail.com,
bruce.poon@ieee.org

² School of Electrical and Information Engineering, University of Sydney,
NSW 2006, Australia
h.yan@cityu.edu.hk

³ Department of Electronic Engineering, City University of Hong Kong,
Hong Kong, China

Abstract. Visually impaired people often have to face difficulty when they try to identify denominations of bank notes. Currently in Bangladesh, there is no system that can easily detect the monetary value of the note. Pattern recognition systems developed over the years are now fast enough to do image matching in real time. This enables us to develop a system able to analyze an input frame and generate the value of the paper-based currency in order to aid the visually impaired in their day-to-day life. The proposed system can recognize Bangladeshi paper currency notes with 89.4% accuracy on white paper background and with 78.4% accuracy tested on a complex background.

Keywords: SIFT · SURF · ORB · Visual Assistance

1 Introduction

There are many visually impaired people around the world especially in the developing world. According to a statistics by World Health Organization [1], the total number of visually impaired people in the world is 285 million. 39 million of these people are blind and 246 million of them are affected by vision related problem. About 90% of the total visually impaired population lives in the developing world and most importantly 82% of them are ages 50 years or more. Detecting the value of bank currency is an important aspect if they are to carry out financial activity like any other people.

There have already been a couple of research works on this subject [2][3]. However they were either too broad in their approaches, or the work had been done with some different bank notes. In this project, we concentrate on Bangladesh Bank notes only, and we use algorithms significantly faster than those used in these two research works.

Scale Invariant Feature Transform (SIFT) [4] and Speeded up Robust Features (SURF) [5] are the two methods that are usually used to perform feature matching

between two unknown images. In situation where we need to produce output quickly we need a better solution. Hence, we go for Oriented FAST and Rotated BRIEF (ORB), a method developed in the lab of Open Source Computer Vision (OPENCV) [6].

ORB is much faster than both SIFT and SURF. SURF has a better run-time than SIFT. Each of these three algorithms has been implemented by engineers at OPENCV. In this project, we have tested each of them in desktop environment running Windows 8.1. The mode of vision is via a web-cam that receives input frame from the outside environment, analyzes the frames and then provides information to the user. If there is a paper currency in the frame, it will provide a value of that currency as well. The module runs in real-time, and has the ability to give textual as well as auditory output. The textual output serves as a confirmation for normal users who can see, whereas the auditory output enables visually impaired user to hear.

2 Implementation

As shown in Figure 1, our system first accepts a photograph of a bank note, applies Oriented FAST and Rotated BRIEF (ORB) algorithm on that image and finally creates a database which contains ORB description of all notes.



Fig. 1. Bangladesh banknote pre-processing steps

2.1 Pre-processing and Feature Extraction

In order to initiate the testing procedure, appropriate databases of relevant descriptors are collected. For that, 'good' samples of Bangladesh bank notes are required. The samples collected were mainly drawn from various sources over the Internet. The main reason was that it was difficult to collect bank notes from the Bangladesh Bank (Central Bangladesh Bank) due to various administrative procedures and the bank was located far from the place of our operation. The second reason was to save the extra hassle of scanning the notes via a scanner.



Fig. 2. 50 taka bill with key points

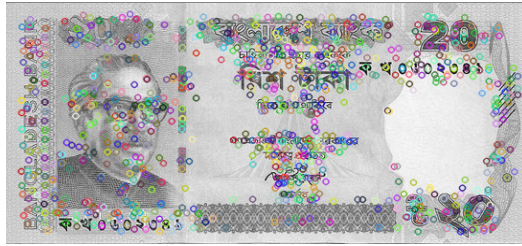


Fig. 3. 20 taka bill with key points

In the above two figures, there are an uncanny amount of similarity between the two bank notes. For a human having normal eye sight, a slight dark environment would confuse the user between the denomination values. It becomes a monumental task for a visually impaired person to differentiate between these two notes.

The bank notes are being pre-processed so that all of them have approximately of the same size, about 500x250 pixels. The descriptor and detector of the notes are found and stored in a table. Both the detectors and the descriptors are found using Oriented BRIEF and Rotated FAST (ORB) [7]. The number of considered features was 500, the number of pyramid level that was used was 8 and the edge threshold was 31[8]. All of these values were used as default when creating the database of descriptors.

2.2 Testing in Real Time

The upper dashed box represents the training of the system and the lower dashed box represents the testing of the system.

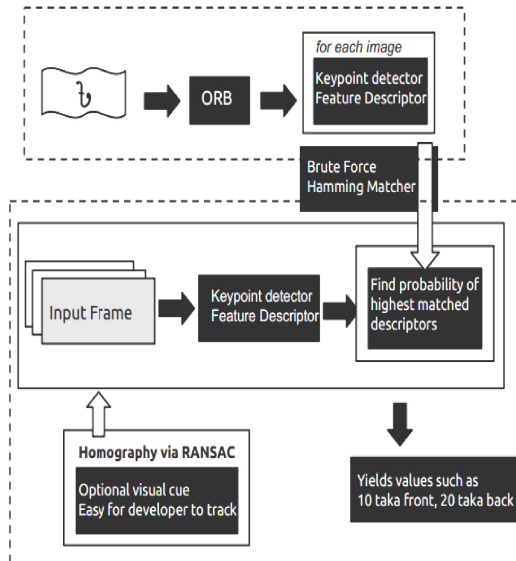


Fig. 4. Overview of the system

Once the descriptors have been computed, they are then stored into a table with the corresponding label of the bank notes. For testing purposes, user input is taken via an HD web-cam. The camera sends about 30 frames per second. On receiving an image, the module computes the detectors found in the image using FAST [9]. Using the detectors, the corresponding descriptors are calculated using BRIEF [10]. However, as BRIEF has little notion of direction embedded in it, it has to be found out using centroid intensity. The next part involves matching of the descriptors of the input frame with that of the training samples.

2.3 Thresholding

The content of the input image often determines the quality of the descriptor that is chosen to represent a good match. In settings where there is no background, the distance between the descriptors will be different from when there is background.

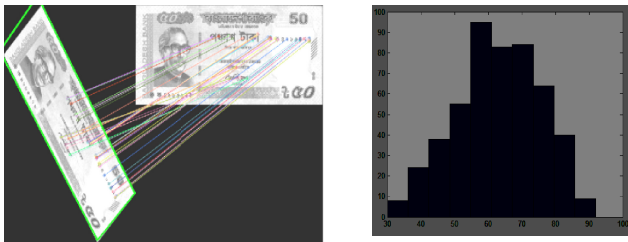


Fig. 5. Money with no background (left) and the distance of the descriptors from training and testing sample (right)

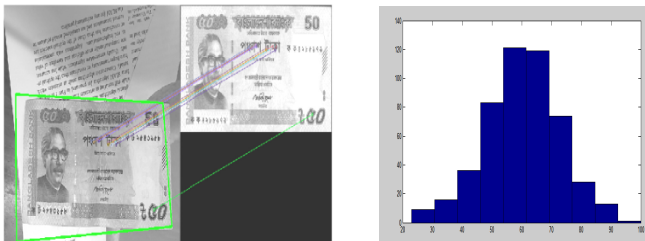


Fig. 6. Money with background (left) and the distance of the descriptors from training and testing sample (right)

In normal situation where there is the presence of background, only descriptors whose distance is less than twice the distance of the minimum distance between the descriptors [11] are used.

$$match_desc_dist < 2 * \min(match_desc_dist)$$

The frequency of the matched descriptors is analyzed. The highest frequency indicates the presence of a given bank note.

2.4 Homography Calculation

A training image of Taka 50 has to undergo rotation along with translation to match with the image in real time. The phenomenon is known as homography and is calculated using:

$$x' = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} \tag{1}$$

$$y' = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} \tag{2}$$

x' and y' are coordinates in the test scene, whereas x and y are coordinates in the training scene. They are bounded to one another in value via the constants a_1 to a_8 . The above two equations are examples of over constrained system which can be solved using either Least Squared Error or via Random Sampling Consensus (RANSAC).

In Least Squared Error, the error between the estimated and the actual output is minimized by a certain iterative procedure. In RANSAC, two points are randomly chosen to fit a line. The error between the estimated and the actual output is found out. If that error is above a threshold then two random points are again chosen. If the error is lower than the threshold, the loop stops.

3 Experimental Results

The system was developed using OpenCV 2.4.6 [6] and mexopencv [12]. As an input medium, Logitech HD web-cam was used. The web-cam was portable to scan two types of currency transformation (full and half folded). The experiment was also conducted using a contrastive background.



Fig. 7. Full and half folded 50 taka note with background (above), full and half folded 20 taka note without background (bottom)

4 Conclusions

This experiment shows that it is possible to detect and enumerate values of an unknown bank note, even if they are very close to another in composition. One of the other goals of this experiment is to extend it to the mobile world as well. Even if it is obvious that it is impossible for every visually impaired people in a developing country to own a smart-phone, it is not that hard for people living in the developed world. Implementing this algorithm to recognize bank currencies for such demography will definitely improve their economic lifestyle. However, on Windows 8.1 platform, the proposed system can only recognize Bangladeshi paper currency notes with 89.4% accuracy on white paper background and with 78.4% accuracy tested on a complex background.

Acknowledgement. This project is supported by a grant from Independent University Bangladesh (IUB). We would like to thank Mr. Forhan Noor for helping us with the experiments.

References

1. WHO Media Centre fact sheet no. 282 (October 2013). <http://www.who.int/mediacentre/factsheets/fs282/en/>
2. Digman, M., Elder, C.: Mobile Banknote Recognition and Conversion. https://stacks.stanford.edu/file/druid:yt916dh6570/Elder_Digman_Foreign_Bill_Recognition.pdf
3. Bruna, A., Farinella, G.M., Guarnera, G.C., Battiato, S.: Forgery Detection and Value Identification of Euro Banknotes. *Sensor (Basel)* **13**(2), 2515–2529 (2013). doi: 10.3390/s1302022515. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3649387/> (February 18, 2013)
4. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* (2004). <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
5. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-Up Robust Features (SURF). ftp://vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf
6. Open Source Computer Vision. <http://opencv.org/>
7. Perception, M., Group, R.: Chubu University, Japan. http://www.vision.cs.chubu.ac.jp/CVR/pdf/Rublee_iccv2011.pdf
8. Open CV, Feature Detection and Description. http://docs.opencv.org/modules/features2d/doc/feature_detection_and_description.html#orb-orb
9. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. *European Conference on Computer Vision* (May 2006). http://www.edwardrosten.com/work/rosten_2006_machine.pdf
10. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part IV*. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
11. Issues with imgIdx in Descriptor Matcher mexopencv, Stack Overflow. <http://stackoverflow.com/questions/20717025/issues-with-imgidx-in-descriptormatcher-mexopencv>
12. Yamaguchi, K.: [GitHub.com/mexopencv](https://github.com/mexopencv). <https://github.com/kyamagu/mexopencv>

Face Recognition Using Genetic Algorithm

Qin Qing and Eric C.C. Tsang^(✉)

The Macau University of Science and Technology, Taipa, Macau
qinqing_alice@163.com, cctsang@must.edu.mo

Abstract. Recently human faces recognition has become a significant problem in many fields especially in criminal investigation area. In order to minimize the scope of searching for a suspect, it is necessary to adopt a method to search the suspect quickly and efficiently. This paper achieves the recognition of human faces by using genetic algorithm. The unique selection of chromosome coding method and the method to select a fitness function are presented. Since human faces include various expressions and different angles of photographs which added to the difficulties of recognition, this article adopts the face, eyes and mouth as the feature extraction which reduces the risk of adverse factors and increases the recognition rate. These three characteristics are fused to make a new face. In the procedure of matching, the foundation to the similarity calculation is the principal component of each feature. Besides, it is the fitness function that measures the characteristics of the suspect and the Euclidean distance between the principal components of each human feature. It implements the value of the fitness of chromosome and accomplishes the automatic recognition.

Keywords: Genetic algorithm · Recognition · PCA

1 Introduction

Nowadays human face recognition becomes more and more important and it has become one of primary subjects in the research areas of image processing. Discriminating between faces is not easy because they contain the same features such as eyes, mouth, nose etc. Small differences could be found in the positions of these features in the faces, the general face shape and in color [1]. Many approaches for face recognition have been developed; Chellappa et al. present in [2] an overview of the different face recognition techniques. Among these techniques, subspaces based methods (such as Eigenfaces [3] and Fisherfaces [4] etc.) have been successfully applied, because these methods allow efficient characterization of a low-dimensional subspaces while preserving the perceptual quality of a very high-dimensional face image. Eigenface method based on PCA is the most popular method [5]. Criminal Investigation using genetic algorithm to achieve a high efficiency is of practical significance. GA identifies the characteristics of main suspects and performs automatic matching. By using this method, the number of identified suspects could be reduced from hundreds to a few. Thus, the police could narrow down the scope of investigating the suspects.

2 GA and PCA Methods

Genetic algorithm was formally introduced in the 1970s by John Holland at University of Michigan [9]. In the computer science field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (sometimes called a meta heuristic) is routinely used to generate useful solutions for optimization and search problems [6]. Genetic algorithm generates solutions to optimization problems using the techniques inspired by natural evolution. These techniques include selection, crossover and mutation, which involved various fields such as computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and others.

Principal component analysis (PCA) is an analysis technique to simplify data sets. Principal component analysis is often used to reduce the dimension of a data set. The data set maintains the largest contribution to the variance of characteristics. This is the main component by keeping low-level, while ignoring the higher-order principal components. Such low-level components are often able to retain the most important aspect of live data. However, it is depending on the specific application. Since the dependency of the principal component analysis is on the data, it will have a great influence on the accuracy of the data analysis.

3 Proposed Method

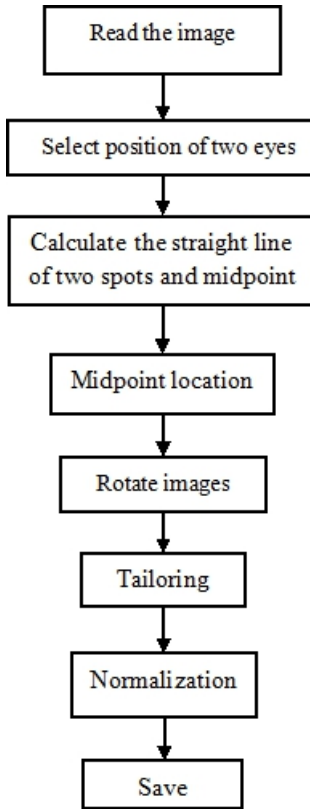
3.1 Pre-processing Database

Since the results of experiment are easily affected by external interference such as facial expressions, illumination, differences of positions etc. The first procedure is to pre-process image database. Figures 1 to 3 represent before and after rendering of locating human eyes and the normalization of size.

Human face images trimming puts all the pictures into a same size and were stored. After pre-processing the size of each image will be unified to the same dimension (128*128 temporarily). The resource of database is obtained from ORL human faces in university of Cambridge. This is a well-known database of faces, which includes a set of face images taken between April 1992 and April 1994 at the lab. The database was used in the context of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department [7].

3.2 Segmentation

After human face images are pre-processed, the next part is to divide eyes and mouth into different area as shown in figure 4. It does the calculation to obtain the principal component analysis of three facial features which are the facial contours, eyes and mouths. Calculated results are saved as the positions of eyes and mouth respectively as well as the PCA of eyes, mouth and faces of database.



Flowchart 1: Pre-processing Steps

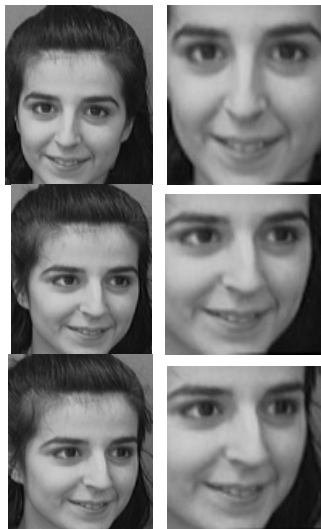


Fig. 1. & 2. Size normalization

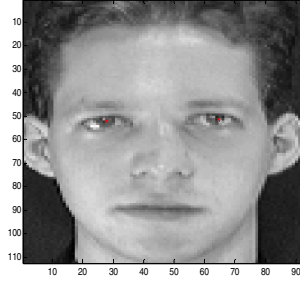


Fig. 3. Locating human eyes

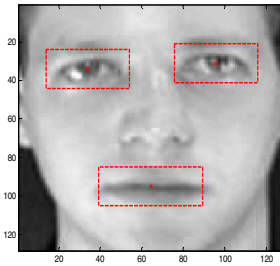


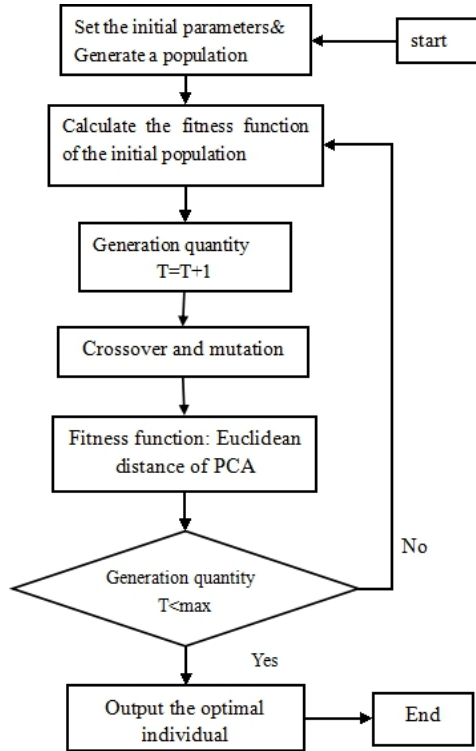
Fig. 4. Dividing eyes and mouth into areas

3.3 Recognition

GA is used to perform recognition automatically. The steps are shown in flowchart 2.

We perform recognition by using genetic algorithm. First of all we need to consider how to code the features of images. Coding is an important step in genetic algorithm and a key to design a good method. Encoding method for GA affects the computation time of crossover, mutation and genetic operators, which largely determines the efficiency of genetic evolution. There are several kinds of coding methods that include binary coding, gray coding and floating point coding etc. Because of the availability of binary coding and decoding and the easiness to realize the selection, crossover and mutation operation, this paper adopts the binary coding. In order to achieve the automatic matching of a suspect, it is necessary to perform feature partition of human faces. Besides, we choose the face, eyes and mouth that represent facial features best as a basis for identification. Thus each chromosome should illustrate the characteristics of three variables: x_1 , x_2 and x_3 which denote the face, eyes and mouth respectively. Each variable accounts for the chromosome length of $1/3$. Moreover, the only one gene showing “1” means that the corresponding characteristics of the person selected from database is the most optimal matching as shown in the following figure (Note: the chromosome length represents the number of people in the image database).

$$\begin{array}{c} | 0 0 0 0 0 0 0 0 0 0 1 | 0 0 0 1 0 0 0 0 0 0 0 0 | 1 0 0 0 0 0 0 0 0 0 0 0 | \\ \hline x_1 \qquad \qquad \qquad x_2 \qquad \qquad \qquad x_3 \end{array} \tag{1}$$



Flowchart 2: Recognition Steps in GA

4 GA Operations

4.1 Fitness Function

After chromosome coding, it needs fitness assessment for each chromosome that determines which chromosome should be selected and copied into the crossover and mutation stages in the process of evolution. The functions, g_1 , g_2 and g_3 represent the Euclidean distance of the human face, eyes and mouth of principal component among the suspects and the image database. It is evident that genetic algorithm implements the automatic matching features by searching for the optimal chromosome and identifying the most approximation of feature matching from database to the suspect.

$$\min f(x_1, x_2, x_3) = \sum_{i=1}^{10} x_1(i) * g_1(i) + \sum_{i=1}^{10} x_2(i) * g_2(i) + \sum_{i=1}^{10} x_3(i) * g_3(i) \quad (2)$$

4.2 Selection of Chromosomes

The selection (reproduction) operator is intended to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation [8]. Based on roulette algorithm, it implements the selection

and duplication. The probability of each chromosome selected to next generation is shown as follows

$$P(i) = fitness(i) / \sum fitness \tag{3}$$

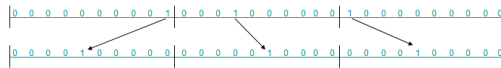
4.3 Mutation

To guarantee diversity of individuals, it is essential that chromosome crossover and mutation be applied to avoid obtaining partial optimal solutions.

The traditional genetic algorithm generally chooses between two individuals of single point crossover using random variation. Namely first generates a random variation of the location of the chromosome, the original 1 of the genetic variation is changed to 0. If the variation of the location of the gene is 0, it will be changed to 1. Moreover, the probability of crossover and mutation is fixed, which is not beneficial to the preservation of elite individuals and reduces the search efficiency of the algorithm.

4.3.1 Mutation of Chromosome Location

When a code is set that the chromosome of each section can only have one “1”, it will occur that the chromosome variation could be both “1” or “0”. So the traditional method of mutation is not applicable. This paper proposes the mutation in the position of chromosome represented by “1”, as shown in the following figure, in which the location of mutation is randomly selected.



4.3.2 Probability of Crossover and Mutation

Under the condition that the probability of crossover and mutation is fixed in the traditional GA, elite chromosome is likely to be changed into a new individual that will cause the decline of searching efficiency. Consequently, we use a sigmoid curve to implement the transformation of fitness. According to the formula, with the higher fitness of a chromosome, the probability to be crossed over and mutated is much smaller. With this method the elite chromosome would get greater chance to be preserved. The following equation represents the self adjustment of the probability of performing cross over and mutation. In our method, we change this probability based on the fitness of a chromosome.

$$P = \begin{cases} \frac{P_{\max} - P_{\min}}{1 + \exp(9.9034(\frac{2(f - f_{avg})}{f_{\max} - f_{avg}}))} + P_{\min}, & f \geq f_{avg} \\ P_{\max} & f < f_{avg} \end{cases} \tag{4}$$

5 Result Analyses

When it comes to the training and testing results, after selection, crossover and mutation, we record the best individuals and the fitness of chromosomes.

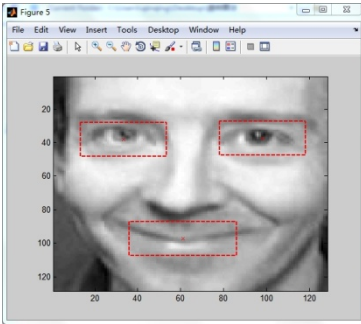


Fig. 5. Original image

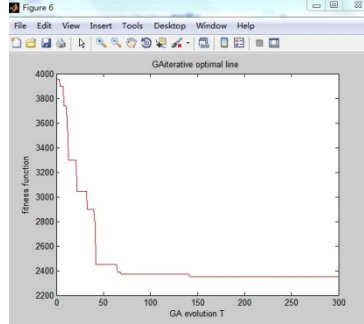


Fig. 6. GA Cycle in recognition

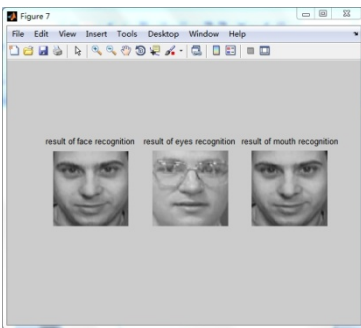


Fig. 7. Identified images

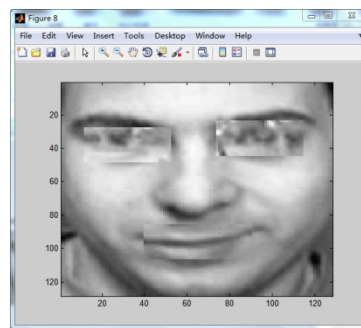


Fig. 8. Combined image

The best result is shown as follows:

Best face individual: 0000000001
 Best eyes individual: 0000001000
 best mouth individual: 0000000001

It can be seen from the recognized chromosomes that genetic algorithm is feasible and useful in solving many face recognition problems.

6 Conclusion

This paper achieves the face recognition by using genetic algorithm. It solves the problem by choosing the faces, eyes and mouth as the characteristics to optimize every part separately, which benefits a lot in criminal investigation. If we pay too much attention to coding, fitness function will be the only goal programming method that only runs a simple weighted calculation, which will bring disadvantages to the results of matching and recognition considerably. In our method there are some drawbacks

existed in the fusion of images. The new faces that consist of segmentations have not been obtained accurately so far.

Acknowledgments. This research work is supported by the Macao Science and Technology Development Fund.

References

1. Kamngam, S., Fukumi, M., Akamatsu, N.: Face Recognition using Genetic Algorithm based Template Matching. In: International Symposium on Communications and Information Technologies 2004 (ISCIT 2004), Sapporo, Japan (October 26–29, 2004)
2. Chellappa, R., Wilson, C.L., Sironhey, S.A.: Human and machine recognition of faces: A survey. *Proceedings of IEEE* **83**(5), 705–740 (1995)
3. Turk, M., Pentland, A.: Face recognition using eigen-faces. In: *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 586–591 (1991)
4. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigen-faces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**(7), 711–720 (1997)
5. Xu, Y.-Q., Li, B.-C., Wang, B.: Face Recognition by Fast Independent Component Analysis and Genetic Algorithm. In: *Proceedings of the Fourth International Conference on Computer and Information Technology (CIT 2004)*. IEEE (2004)
6. http://en.wikipedia.org/wiki/Covariance_matrix
7. <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
8. Arum, S., Harish, G., Salomon, K., Saravanan, R., Kalpana, K., Dr. Jaya, J.: Neural networks and genetic algorithm based intelligent robot for face recognition and obstacle avoidance. IEEE 2013 IEEE-32107 Coimbatore, India (July 3, 2013)
9. Mitchell, M., (ed.): *An introduction to Genetic Algorithms*. MIT Press

Face Liveness Detection by Brightness Difference

Patrick P.K. Chan and Ying Shu^(✉)

School of Computer Science and Engineering, South China University of Technology,
Guangzhou 510006, China
patrickchan@ieee.org, yingsy13579@gmail.com

Abstract. This paper proposes a method to detect face liveness against video replay attack. The live persons are distinguished from and video replay attack by analyzing the brightness difference on the face and background. By taking photos with/without a flashlight, the brightness differences of the face are compared with the one of the background. The live person and the attack should have different brightness differences. The accuracy on the liveness detection using the proposed model is satisfying in the experiments.

Keywords: Face liveness detection · Brightness difference · Flashlight

1 Introduction

In recent years, person biometric identification has been widely used in security surveillance due to its satisfying performance. The most well-known techniques include fingerprint [16], iris [17] and facial recognition [1]. However, an adversary who intentionally downgrades the performance of the system may exist in these security applications. For instance, a spoofing attack [2] presents a copy of biometric traits of a legal user to spoof the person identification system. In facial recognition, the biometric traits are the facial photograph and video [9], which can be obtained easily nowadays because of the rapid development of hardware (e.g. a high quality camera and screen in a smart phone). As a result, the robustness of facial biometric identification is an important research topic recently [18].

Liveness facial detection has been proposed in order to recognize whether the object is a real person. The detection methods can be separated into two categories according to whether an additional device is required. One example of the methods without additional device is to detect spontaneous eye blinks [3]. Eye blinks is an essential motion of a live person. However, this method only applicable to defense against the photograph attack but not the video attack as the eye blink can be recorded in a video. As the textures of a real human face are different from a photograph or a screen, this information has been applied to liveness detection. The examples are Uniform Local Binary Patterns (LBP) [4] and texture features from Gray-Level Co-occurrence Matrix (GLCM) [5]. One drawback of these methods is large computational complexity since each frame should be calculated by temporal processing strategies [6]. It has also been found that a live face has subtle changes like the change of color and movement due to

the blood flow [7]. These changes are magnified by Eulerian magnification [7] which also increase the time complexity. Reflectance disparity between real faces and fake materials [8] is a method with additional device. Wave signals with different lengths are emitted to forehead region of the object. Facial skin and other materials have different albedo. Although this method is 97.78% accurate, it requires special IR (infrared ray) LEDs of 685 and 850nm wavelengths and the angle between LEDs and camera must strictly be 45°. Its implementation cost is relatively high.

In this paper, we investigate the video replay attack, which play back the video of a user in a tablet in front of the camera of facial recognition system. We propose a method which calculates the brightness difference between the background and the person under a flash. If the object is a real person, the difference is larger due to the distance between flashlight and background is larger than the one with the person. On the other hand, the background and the human displayed in a photo and video should have similar brightness since both of them are displayed on a tablet. This method has the advantages of methods with (i.e. high accuracy) and without additional device (i.e. low implementation cost) by installing a low-cost and simple flashlight in the system. The experimental results show that our proposed method has a satisfying result.

The rest of the paper is organized as follows. A brief review of relevant works is given in section 2. Section 3 discusses the motivation and the proposed method is devised in section 4. The experimental results are discussed in section 5. Finally the conclusion is given in section 6.

2 Related Work

A person attempts to access the system by pretending a legal user in a spoofing attack. Most widely used spoofing attacks of face recognition are photograph attack, video replay attack and fake face attack[1]. Video replay attack and photograph attack are also 2D face spoofing attacks (i.e., pretending by a planar objects, e.g. photograph). They present a photo or a video which has the biometric traits of a legal user to spoof the detection system. The video replay attack provides dynamic biometric (i.e., motion of a user) traits while the static traits are provided by photograph attack. Differently, fake face attack is 3D face spoofing attack method. Attackers make a mask or clay face to spoof system. Fake face can present 3D biometric traits of face. 3D face spoofing attack cost high but it is difficult to detection. We focus on 2D video replay attack problem.

Many countermeasures for face anti-spoofing have been proposed. Eye blinks detection [3], which captures human blink, defenses against the live face and photograph with satisfying result. However, its performance drops on the situations of wearing glasses and video replay attack. Another liveness detection method is Optical flow [10], which detects the degree differences of reference field and actual optical flow filed. It relies on precise computation of optical flow filed and the illumination affects the accuracy significantly. Uniform Local Binary Patterns (LBP) [4] and texture features from Gray-Level Co-occurrence Matrix (GLCM) [5] representing static texture of a face are applied in face liveness detection. Dynamic texture content [6, 11] is analyzed by the temporal processing strategies also achieve a good performance. Unfortunately, these methods have a high time complexity.

The relative movement intensity between the face and the scene background is also a counter measure of photo attack [12]. This method measures the tiny movement of a human. It is good at detecting the paper-based print attacks but not for video or 3D mask attack. By adding different wavelength illumination to forehead region [8], the skin and mask has different albedo. Measuring reflectance disparity can be used in face liveness detection. However, it needs special IR (infrared ray) LEDs, which is a strict requirement.

3 Motivation

This section discusses the motivation of using a flashlight in liveness detection by using a simple example.

Figure 1 shows an example of a live person and spoofing video with and without flashlight. When the flashlight is applied, a live person (figure 1b) can be easily distinguished from a spoofing video (figure 1d). The brightness between the face is large but the background is small when comparing the images with and without the flashlight for a live person. Differently, a big light spot is located at the center of image with flashlight for a spoofing video. Moreover, the brightness of the rest of the image is similar to the one without flashlight. This observation motivates us to consider the brightness difference the face and background separately for liveness detection.

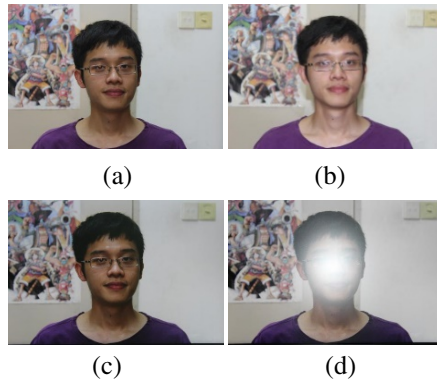


Fig. 1. Examples of live person and spoofing video with / without flashlight. (a) Live person without flashlight (b) Live person under flashlight (c) Video screen without flashlight (d) Video screen under flashlight.

4 Proposed Method

In this paper, a liveness face detection method which calculates the brightness difference the face and background with flashlight is proposed. Two images using and without using the flashlight are taken from the object. The brightness values of the face and the background are extracted from the images. The differences of the brightness values of face and background are calculated separately as the input features for the liveness face detection. Section 4.1 discuss the procedure of face and background identification while the calculation of the brightness value is mentioned in section 4.2.

4.1 Face and Background Identification

The face location process proposed in [13] has been applied. The image is divided into 3×3 blocks as local areas. For each local area, SMQT transformation [19] is applied to enhance the details of structural information and reduce the sensitivity to illumination. The enhanced information is input to a split up SNoW classifier [20], which detects faces with different features, expressions, and poses under different lighting conditions. This face detection has satisfying results in the two commonly used upright face detection database: BioID and CMU+MIT. Figure 2a shows the identified face region.

A simple ad-hoc algorithm is applied to identify the area of the background. Two rectangles are located on the top-right and left corners. The width is 200 pixels which is determined according to the results of the experiments. The height of the rectangle is determined according to the location of the face. Thus, the region of the background does not cover the shoulder. The background areas are illustrated in figure 2b.

As mentioned previously, a big light spot is located at the center of image with flashlight for a spooking video, shown in figure 2c. It causes the face location inaccurate. As a result, the face and the background region for an image with the flashlight follow the ones without the flashlight.

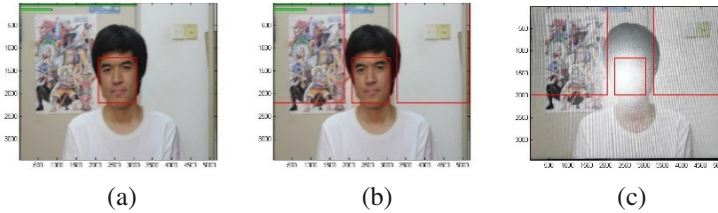


Fig. 2. Examples of face and background identification. (a) Region of the face identified by SMQT and SNoW (b) Region of the background identified based on the face region (c) Region of face and background identified based on the (b).

4.2 Brightness Value Calculation

A color image contains a number of pixels which contain the values of RGB (Red, Green and Blue) with the range $[0, 255]$. The brightness values of a pixel can be represented by its gray value [14]. So we need to calculate the gray value to learn the brightness value. There are three methods of image gray processing [14] to calculate gray value. Average, weighted average and maximum value methods are shown in (1), (2) and (3) respectively.

$$F(i, j) = (R(i, j) + G(i, j) + B(i, j))/3 \quad (1)$$

$$F(i, j) = 0.2989R(i, j) + 0.5870G(i, j) + 0.1140B(i, j) \quad (2)$$

$$F(i, j) = \max(R(i, j), G(i, j), B(i, j)) \quad (3)$$

Where i and j are the coordinate of a pixel, and the function F , R , G and B are the gray, red, green and blue values of a pixel. The coefficient values in the weighted average method are suggested in [14]. The gray values have the same distribution and characteristic of chroma and brightness of the color image. In this paper, equation (2) is applied.

The difference of brightness between the face (Diff_{face}) and the background (Diff_{back}) with and without the flashlight is defined as:

$$\text{Diff}_R = \mathbb{E}_{(i,j) \in R} (F_{NoFL}(i,j)) - \mathbb{E}_{(i,j) \in R} (F_{FL}(i,j)) \quad (4)$$

Where R is the set of pixels in the face (*face*) or the background region (*back*), and F_{status} is the gray value with and without the flash light (*FL* or *NoFL*).

5 Experiment

5.1 Dataset Generation

A dataset is collected by using a digital single lens reflex (DSLR) camera with the model of the DSLR is Canon EOS 600D. We define the positive class contains malicious samples (spoofing attack) and the negative class contains legitimate samples (live person). 12 males and 9 females with the ages from the age 19 to 22 are invited as the object. For each object, two photos are taken on the live person with and without flashlight to generate the negative sample. Then the object's photo is displayed on a tablet to simulate the spoofing attack. Another two photos are taken on the tablet displaying the person's image with and without flashlight to generate the positive sample. As a result, 21 samples of each class are collected. Each experiment has been repeated 10 times independently.

5.2 Accuracy

The testing accuracy of the proposed method is evaluated in this section. The dataset is spitted into half randomly as training and testing set. The classifiers including SVM with the linear kernel (SVM-Linear), Multi-Layer Perceptron Network (MLP), K-Nearest Neighborhood (k-nn), Bayesian classifier (Bayes) and Radial-Based Function Network (RBF) and Decision Tree (DT) are applied. Their parameters are determined according to 5-fold cross validation. The experiment has been executed 10 times independently.

We firstly investigate show the brightness and its difference values of the face and the background for a live person and replay video attack, reported in table 1. Without the flashlight, the differences between the brightness of the face and the background for a live person and a replay video are similar. However, the brightness of the face of the spoofing attack increases significantly in comparison with the one of the live person. On the other hand, the increase of the brightness of the background for the live person due to the flashlight is more than the one for the spoofing attack. Therefore, the proposed features are useful to distinguish a live person from a spoofing attack.

The testing accuracy of different classifiers using the brightness difference values are reported in Table 2. Generally, all classifiers using the proposed features achieve a good result. SVM-linear, K-nn, Bayes and Decision Tree have achieved 100% accurate while MLP classifier is 97.50% and the RBF is 95%. The experimental results suggest that the proposed methods detect the spoofing attack efficiently.

Table 1. Examples of brightness and the difference values between the situation with and without flashlight

Examples		Face		Difference	Background		Difference
1	Live person	No FL	101.80	35.07	No FL	167.17	7.52
		FL	136.87		FL	174.69	
	Replay attack	No FL	108.31	104.89	No FL	164.26	-1.49
		FL	213.20		FL	162.77	
2	Live person	No FL	94.40	37.17	No FL	174.92	14.46
		FL	131.57		FL	189.38	
	Replay attack	No FL	84.03	130.20	No FL	158.80	8.55
		FL	214.23		FL	167.35	
3	Live person	No FL	100.20	35.32	No FL	177.91	11.00
		FL	135.52		FL	188.91	
	Replay attack	No FL	88.83	123.97	No FL	161.49	9.31
		FL	212.80		FL	170.80	

Table 2. Accuracy of the proposed method using different classifiers

Classifier	SVM-Linear	MLP	RBF	k-nn	Bayes	DT
Accuracy	100%	97.50%	95.00%	100%	100%	100%

6 Conclusion

A method of liveness face detection considering the brightness of a face and a background by adding a flashlight in the system is proposed. A flashlight increases the difference on the brightness between the face and the background for a live person and a spoofing video attack. The experimental results show that the brightness of a face increases significant for a spoofing attack than the ones of a live person. By using the proposed futures, the well-known classifiers have satisfying performance.

Acknowledgements. This work is supported by a National Natural Science Foundation of China (61272201), and a Fundamental Research Funds for the Central Universities (10561201465).

References

1. Ross, A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics, vol. 6. Springer (2006)
2. Schuckers, S.: Spoofing and Anti-Spoofing Measures. Information Security Technical Report 7(4), 56–62 (2002)

3. Pan, G., Sun, L., Wu, Z., Lao, S.: Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcam. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
4. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **4**(7), 971–987 (2002)
5. Haralick, R., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics* **3**(6), 610–621 (1973)
6. de Pereira, T.F., Anjos, A., De Martino, J.M., Marcel, S.: LBP-TOP Based Countermeasure against Face Spoofing Attacks. *Computer Vision with Local Binary Pattern Variants-ACCV*, pp. 121–132 (2012)
7. Wu, H.-Y., Rubinstein, M., Shih, E., Gutttag, J., Durand, F., Freeman. Eulerian, W.T.: Video Magnification for Revealing Subtle Changes in the World. *ACM Transactions on Graphics* **31**(4) (2012)
8. Zhang, Z., Yi, D., Lei, Z., Li, S.Z.: Face Liveness Detection by Learning Multispectral Reflectance Distributions. In: *IEEE Automatic Face & Gesture Recognition and Workshops*, pp. 436–441 (2011)
9. Chakka, M.M., Anjos, A., Marcel, S., Tronci, R., Muntoni, D., Fadda, G., Pili, M., Sirena, N., Murgia, G., Ristori, M., Roli, F., Yan, J., Yi, D., Lei, Z., Zhang, Z., Li, Z.S., Schwartz, W.R., Rocha, A., Pedrini, H., Navarro, L.J., Santana, C.-M., Määttä, J., Hadid, A., Pietikäinen, M.: Competition on Counter Measures to 2-D Facial Spoofing Attacks. In: *IEEE International Joint Conference on Biometrics*, pp. 1–6 (2011)
10. Bao, W., Li, H., Li, N., Jiang, W.: A Liveness Detection Method for Face Recognition Based on Optical Flow Field, Image Analysis and Signal Processing, pp. 233–236 (2009)
11. Komulainen, J., Hadid, A., Pietikäinen, M.: Face Spoofing Detection Using Dynamic Texture. In: Park, J.-I., Kim, J. (eds.) *ACCV Workshops 2012, Part I. LNCS*, vol. 7728, pp. 146–157. Springer, Heidelberg (2013)
12. Anjos, A., Marcel, S.: Counter-Measures to Photo Attacks in Face Recognition: A Public Database and a Baseline. In: *IEEE International Joint Conference on Biometrics*, pp. 1–7 (2011)
13. Nilsson, M., Nordberg, J., Claesson, I.: Face Detection Using Local SMQT Features and Split upSnow Classifier. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 589–592 (2007)
14. Rafael, C.; Gonzalez. Richard Woods. *Digital Image Processing*, Prentice Hall PTR (2002)
15. Li, J., Wang, Y., Tan, T., Jain, A.K.: Live Face Detection Based on the Analysis of Fourier Spectra, *Defense and Security. International Society for Optics and Photonics*, pp. 296–303 (2004)
16. Marcialis, G.L., Lewicke, A., Tan, B., Coli, P., Grimberg, D., Congiu, A., Tidu, A., Roli, F., Schuckers, S.: First International Fingerprint Liveness Detection Competition—LivDet 2009. In: Foggia, P., Sansone, C., Vento, M. (eds.) *ICIAP 2009. LNCS*, vol. 5716, pp. 12–23. Springer, Heidelberg (2009)
17. Toth, B.: Biometric Liveness Detection. *Information Security. Bulletin* **10**(8), 291–297 (2005)
18. Jain, A., Hong, L., Pankanti, S.: Biometric Identification. *Communication of ACM* **43**(2), 90–98 (2000)
19. Nilsson, M., Mattias D., Ingvar, C.: The Successive Mean Quantization Transform. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* 4, pp. 429–432 (2005)
20. Yang, M.-H., Roth, D., Ahuja, N.: A Snow-Based Face Detector. *Neural Information Processing System* **12**, 855–851 (2000)

Sampling and Big Data

User Behavior Research Based on Big Data

Suxiang Zhang^{1(✉)} and Suxian Zhang²

¹ State Grid Information and Telecommunication Co. Ltd., Beijing 100761, China
zsuxiang@163.com

² Department of Foreign Language of Hebei University, Baoding 071000, China

Abstract. In this paper, the enterprise user behavior had been studied based on big data. By combining cloudy computing and k-means clustering algorithm, we proposed the parallel k-mean clustering. The feature were chosen as follows: Power consumption rate in the peak load time; the load rate and the power consumption rate in the valley time and so on. The feature weight can be calculated with entropy weight method. The experimental data came from the intelligent industrial park of Gansu province. The enterprise users are classified into two classes, the different type enterprise has their electricity law. In the future, enterprise can optimize their working time, lower the electricity cost in the same power consumption. This provides strong support for the demand of side response of power grid.

Keywords: Big data · Parallel K-means · Cloudy computing · User classification

1 Introduction

With the increasing pressure of resources and environment, the demand for environment protection, energy conservation and emissions reduction and sustainable development requirements had been more and more eager. As a new type of smart service network, the smart grid will effectively integrate data of the power system, optimize the operation of the power infrastructure and management, and promote interaction with users by using an open information system and the information sharing model. Based on this background, many countries and regions had regarded the smart grid as an important development goal of power grid in the future, and made a wide range of research and practice for the smart grid. The energy efficiency management had become the main research target in the power consumption of each country.

In recent years, with the development of smart grid construction, the state grid company had built the intelligent industrial parks in Baiyin of Gansu, Dongying of Shandong and Nanjing of Jiangsu, which were focused on the monitoring and management of energy efficiency for big user. Through smart power consumption, a large amount of data was used to study power consumption behavior of user, decision analysis and energy management were provided for the security of power grid in the future.

Some studies had been developed for power users classification, for example, based on the traditional industry, user classification based on clustering was proposed in [1],

the classification results are similar to the directory electricity price in accordance with industry classification results, but it does not take into account the difference with the way the power consumption is calculated. In reference[2], some market values were proposed which included the current market value, the potential market value and regional contribution value, and for the different types of users. Different marketing strategy were formulated, but this class division is very macroscopic, energy does not conform to the fine management strategy of the electrical load. Fuzzy c-means clustering method was proposed for substation power load [3], the substation load could be classified into the industrial, agricultural, municipal and other categories, which concluded that this method is superior to clustering method based on equivalence relation. Other research of electricity prices had not involved user classification [4-5].

In this paper, enterprise user classification was studied based on the clustering algorithm. Clustering is one of the important research topics in data mining in which physical object can be classified into multiple classes or clusters [6]. In the same cluster, objects are as similar as possible, and in the different cluster, objects are as different as possible. Clustering can handle the different field types, find the ability of arbitrary clusters shape, and handle abnormal data, which has some priorities as follows: The data sequence is not sensitive and weak dependence on domain knowledge, etc. However, with the development of the Internet popularization and applications of computer technologies, the demand is becoming more and more for big data clustering analysis. To reduce the processing time or data storage of the clustering algorithm, many methods such as order method, classification method, sampling method, data summary method, parallel and distributed method etc. had been proposed. Common sequence clustering algorithms such as sequential k-means and competitive learning etc., in which all the data do not need to load into memory before clustering, but load data and cluster calculation will be working in the same time. So under the limited computing ability and memory, memory problems had been better solved when dealing with big data. Classification method such as CURE algorithm [8] where the data are divided into disjoint subsets, then the subsets are clustered. The combined clustering result can be regarded as the clustering results of the original data set. Sampling method such as CLARA clustering [9] is assumed that part of the big data sampling can be used to represent the overall clustering results based on the part of the sampling which is approximately regarded as global clustering results. BIRCH [10] as a representative method of summary data, firstly uses the sequential method, then gets the summary information and finishes the clustering algorithm. Map Reduce distributed platforms [11] as a representative of cloud computing, use parallel computing and distributed processing in multiple processors systems. In front of the huge amounts of electricity data of enterprise users, in this paper we combined the parallel computing of the cloud platform with clustering k-means algorithm. The enterprise user behavior analysis was quickly and efficiently performed. The research was focused on the data size, the algorithm complexity and clustering accuracy.

The contribution of this paper is listed as follows. Firstly, we studied the enterprise user behavior in power industry. Secondly, cloud computing technology was used in smart power consumption fields in the first time. Thirdly, some features were built, which included the peak load power consumption rate, load rate, valley load coefficient

etc. The features play important roles for mining the user behavior. In the future, according to different types of enterprises user, Grid Company will develop different service levels and power demand response measures. With these measures, the shift peak to valley, energy conservation and emissions reduction can be realized.

2 MapReduce

When facing with the big data, we know that the traditional clustering method cannot satisfy the accuracy and speed requirements. We need to find a new way to satisfy big data requirements in the smart power consumption. In this paper, we will use Mapreduce technology as the cloud computing platform,

MapReduce is parallelizable computing software architecture proposed by Google. It can solve the problem of parallel computing of big data, and then the result can be put into the file system or a database.

"Map" step: the input data is read by the master node, which will be segmented into the small data blocks, and used the same method to solve the problem. Then the small blocks of data will be distributed to different work nodes. Each work node cycle does the same thing, a tree structure is generated. Each leaf node has to deal with each specific block of data, and send the results back to the parent node.

"Reduce" step: the processing results of all child nodes are transmitted to main nodes, and then all the results will be integrated, the final results will be sent back to the output.

3 Clustering Algorithm

The whole clustering procedure is expressed as shown in Figure 1.

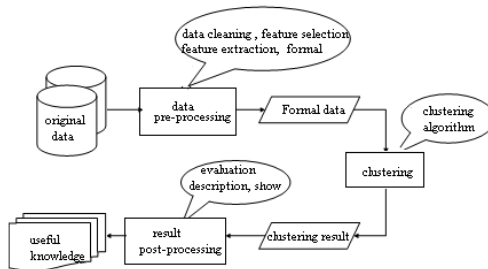


Fig. 1. Clustering procedure

3.1 The Traditional K-Means Algorithm

Its main idea is: k as a parameter of the algorithm, n objects will be classified into k clusters, the objects have high similarity in the same cluster and low similarity between clusters.

① k instances were freely chosen from the data set $\{x_i\}_{i=1}^N$, which are assigned to the initial clustering center $\mu_1, \mu_2, \dots, \mu_k$, N is the sample size.

② The Euclidean distance between the i th instance x_i and every clustering center were calculated, and obtain category of the instance x_i .

$$\mu_j(i) \leftarrow \operatorname{argmin}_i \|x_i - \mu_j\|^2 \tag{1}$$

$$i = 1, \dots, N; j = 1, \dots, k$$

$\mu_j(i)$: aims to the j th cluster which is the nearest distance instance x_i compared with k clusters.

③ According to the equation (2) to recalculate the k cluster centers:

$$\mu_j = \frac{1}{N_j} \sum_{x_i \in \mu_j} x_i, j = 1, 2, \dots, k \tag{2}$$

N_j is the number of objects in the μ_j cluster.

④ Repeat steps ② and ③ until it achieves convergence criterion function. Convergence judgment basis is the square error rule as shown in equation (3).

$$E = \sum_{i=1}^k \sum_{\mu_i} |x - m_i|^2 \tag{3}$$

E is the sum of square error of all the objects in the database, x is the data point in space, m_i is the average of the μ_i cluster, E can make clusters independent as possible.

3.2 Parallel K-Means Algorithm

Parallel k-means algorithm will be realized as shown in Figure 2. The data set can be divided into several data subsets, based on the normal k-means algorithm, the clustering can be realized for the local data sets, and finally the global cluster set can be generated using several local cluster set.

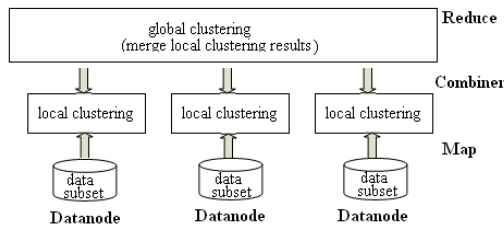


Fig. 2. The parallel clustering model

During clustering, the Datanode can read local data sets in the map phase, output each point and its corresponding cluster. The points containing the same cluster operation will be reduced with combiner operator, the global cluster set can be obtained, and written to the HDFS.

3.3 Features Selection and Weight Calculation

In high dimension, multiple complex, rich noise conditions, the time sequence clustering mining algorithm becomes a research hotspot. And clustering is to search similarity degree of the different objects in the data set, find the similarity between the data. So the research emphasis is to provide a novel, feasible and higher credibility similarity measure for clustering algorithm.

Based on the collected data set, the features were selected as follows:

(1)Power consumption rate in the peak load time everyday: Power consumption amount in the electric power tension time can be divided by power consumption amount in all day

(2)The load rate: The average load will be divided by the maximum load;

(3)Power consumption rate in the low load time everyday;

(4)Power consumption rate in the flat time everyday.

According to the above characteristics, each appraisal object will be represented as a 1×96 row vector matrix: $X = [x_1 \ x_2 \ \dots \ x_{96}]$

And, the standard normalized processing will be carried on for all matrix elements.

$$r_i = [x_i - \min(x_i)] / [\max(x_i) - \min(x_i)], \quad (4)$$

$$r_i \in [0,1], \quad i=1, \dots, 96.$$

For the different feature, the weight value can be calculated based on the entropy method, the entropy values of the feature are shown in equation (5):

$$P = [p_1 \ p_2 \ \dots \ p_i \ \dots \ p_n], \quad (5)$$

$$p_i = \frac{r_i}{\sum_{i=1}^n r_i}$$

Moreover, the weight matrix can be obtained:

$$W = [w_1 \ w_2 \ \dots \ w_i \ \dots \ w_n] \quad (6)$$

$$w_i = 1 - \frac{1}{\ln n} \sum_{i=1}^m p_i \ln p_i$$

The equation (6) will be standardized and the standard state matrix is obtained. In this paper the matrix data classification task can be realized.

4 Enterprise User Electricity Behavior Analysis Based on the Big Data

Load curve is a visual representation for energy consumption behavior of a user, which is the data source of the load analysis. According to the similarity of the user distribution load, the users can be classified, and the targeted operation strategy can be built.

4.1 Data Resource

In smart industrial park of Gansu, there are more than 20 companies. Using the data acquisition system of the enterprise; we obtain electricity data for 24 hours everyday. The sampling frequency is 5 s/times.

4.2 Experimental Results

Based on the big electricity data and parallel clustering algorithm, the enterprise power consumption law can be obtained as shown in Figure 3. From the result, we obtain two laws of the power consumption. The power grid can be regarded as enterprise users A and B types. With analysis of power consumption behavior of the each user, we know that the electricity price is beneficial to A-class enterprise and against to B-class enterprise.

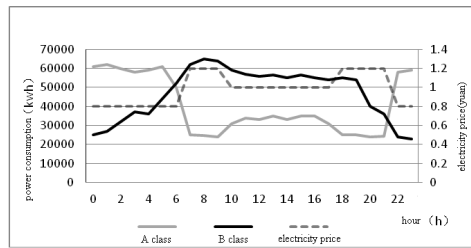


Fig. 3. The different electricity law with the different type enterprise

By using parallel k-means algorithm, we obtained the electric power load characteristics of user in the smart industrial park of Gansu province, that is A type: electricity Price is beneficial to A class enterprise to save electricity fee; B type: electricity price is against B class enterprise to save electricity fee. So we suggested that B type enterprise can adjust their working time to the electricity price which is lower at night. With load analysis, future enterprise can optimize the working time, makes the enterprise in the same power consumption with the lowest electricity cost, and reduce electricity costs. This also provides a strong support for the demand of side response of power grid.

In the future, based on the different user types, power grid can provide different priorities services, which can not only realize the economic profit and social responsibility of Power Company, but also ensure whole power system stability and economic development. At the same time, the users themselves can do shift peak electricity and cost savings, complete the task of energy saving and emission reduction.

5 Conclusions

In this paper, based on big data of the intelligent industrial park, we studied the enterprise electricity behavior analysis model. By combining cloud computing with k-means algorithm, we proposed the multiple power characteristics, with based-entropy weight method, the feature weight can be calculated. Thus the user classification task was completed. The experimental results show that the proposed power user classification method is effective when compared with previous methods. The algorithm was applied in the electric user classification in the first time. Next the demand for side response will be studied based on the different user groups.

Acknowledgements. This work was supported by National High-tech R&D Program of China (863 Program) (No. 2011AA05A116), National major projects of science and technology: Collaborative network control platform and the key technology of ubiquitous networks under multi-terminal (2011ZX03005-004-01); the National Basic Research Program of China (973 Program): Basic theory and practice research of Internet of Things (2011CB302900). The work was also supported by science and technology project of State Grid Corporation (Smart power consumption empirical study)

References

1. Feng, X., Zhang, T.: Research on Electricity Users Classification Technology Based on Actual Load Curve. *Electric Power Science and Engineering* **26**(9), 18–22 (2010)
2. Wang, C., Feng, Q.: The Research of Power Customers Classification Based on Value Assessment. *Value Engineering* **5**, 64–66 (2009)
3. Li, P., Li, X., Chen, H., et al.: The characteristics classification and synthesis of power load based on fuzzy clustering. In: *Proceedings of the CSEE*, vol. 25(24), pp.73–78 (2005)
4. Ruan, W., Wang, B., Li, Y., et al.: Customer response behavior in time-of-use price. *Power System Technology* **36**(7), 86–92 (2012)
5. Zeng, M.: *Study on Time-of-Use Price of Different Customer Classes*, the North China electric Power University (2006)
6. Jun, X., Huang, Y., Li, F.: Research on comparing the sequential learning with batch learning for K-Means. *Computer. Science* **31**(6), 156–158 (2004)
7. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: *Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining*. The Association for the Advancement of Artificial Intelligence, New York, pp. 239–241 (1998)
8. Sun, Y., Li, J., Li, J.: Network users behavior analysis based on CURE algorithm. *Computer Technology and Development* **21**(9), 35–38 (2011)
9. Zhao, G., Guoqing, Q.: Analysis and implementation of CLARA algorithm on clustering. *Journal of Shandong University of Technology: Sci. & Tech.* **2**, 45–48 (2006)
10. Zhao, Y., Guo, J., Zheng, L., et al.: Improved BIRCH hierarchical clustering algorithm. *Computer Science* **3**, 180–182 (2008)
11. Wang, Y., Da, X.: Investigation of distributed and parallel mining calculating architecture and algorithms. *Microelectronics & Computer* **23**(9), 223–225 (2006)
12. <http://en.wikipedia.org/wiki/MapReduce>

13. Zhu, Z., Zhongjian, G., Jinlong, W., et al.: Application of cloud computing in electric power system data recovery. *Power System Technology* **36**(9), 44–50 (2012)
14. Lianshun, M., Cui, L., An, N.: Research and practice of cloud computing center for power system. *Power System Technology* **35**(6), 171–175 (2011)
15. Li, Y., Zhou, X., Wu, Z.: Personal computer cluster based parallel algorithms for power system electromechanical transient stability simulation. *Power System Technology* **27**(11), 6–12 (2003)
16. Liu, M., Chu, X., Zhang, W., et al.: Design of cloud computing architecture for distributed load control. *Power System Technology* **36**(8), 140–144 (2012)

Stochastic Sensitivity Oversampling Technique for Imbalanced Data

Tongwen Rong^(✉), Huachang Gong, and Wing W.Y. Ng

Machine Learning and Cybernetics Research Center
School of Computer Science and Engineering,
South China University of Technology, Guangzhou 510006, China
rongtw2009@gmail.com, wingng@ieee.org

Abstract. Data level technique is proved to be effective in imbalance learning. The SMOTE is a famous oversampling technique generating synthetic minority samples by linear interpolation between adjacent minorities. However, it becomes inefficiency for datasets with sparse distributions. In this paper, we propose the Stochastic Sensitivity Oversampling (SSO) which generates synthetic samples following Gaussian distributions in the Q-union of minority samples. The Q-union is the union of Q-neighborhoods (hypercubes centered at minority samples) and such that new samples are synthesized around minority samples. Experimental results show that the proposed algorithm performs well on most of datasets, especially those with a sparse distribution.

Keywords: Imbalance data · Oversampling · Smote · Q-union · Stochastic sensitivity measure

1 Introduction

With the rapidly development of storage and Internet technologies, the volume of raw data grows explosively and imbalanced learning problem has attracted more and more attentions [1]. In many applications, e.g. financial, medical and other areas related to the national economy and people livelihood, or cybersecurity[2], samples in the minority class can be more important than samples in the majority class. Identification of cancer patients is one of the classical examples of imbalanced problems. From experience, the number of patients without cancer is much larger than the number of patients with cancer. In this imbalanced case, pattern recognition algorithms tend to classify all patients to be "without cancer" to achieve a high average classification accuracy. Therefore, imbalanced pattern classification problems are important and need particular technique to achieve a high classification accuracy for both minority and majority classes.

In imbalance learning, classifiers such as neural networks, SVM, k-nearest neighbors, Bayesian networks, etc., are not designed to deal with imbalanced distributions among samples in different classes. The poor performance of them for imbalanced problem is caused by the presence of under-represented data and severe class distribution skews. When the number of samples in the minority class is heavily outnumbered

by those in the majority class, the minority class is under-represented and usually ignored in learning. Another key factor of classification deterioration is class distribution skews, which may be caused by outliers and noise. The last important factor deteriorating the classification performance is the inappropriate assessment metric. Many traditional classifiers are average accuracy oriented. The average accuracy is defined as the proportion of samples classified correctly to all samples in the dataset. This metric may be inappropriate for imbalanced learning. Assume that a dataset contains of 1% of minority samples and 99% of majority samples. A classifier classifying all samples to be majorities yields 99% accuracy. It is a high accuracy; however this classifier is useless in classifying minority samples from majority samples.

To deal with imbalanced data, sampling methods including oversampling and under-sampling methods are proposed to achieve a relatively balanced distribution to improve performance of standard classifiers. The SMOTE [3] is one of them most widely cited and applied oversampling methods. However, it performs worse when the dataset is sparse. Hence, we propose the Stochastic Sensitivity Oversampling (SSO) which is a new improvement to the SMOTE to enhance its performance, especially for sparse datasets.

Section 2 introduces current under-sampling and oversampling methods. We propose the SSO in Section 3. Experimental results and discussion are presented in Section 4. We conclude this work in Section 5.

2 Related Works

Undersampling and Oversampling are two major techniques for dealing with imbalanced pattern classification problems. Current undersampling methods, oversampling methods and the sensitivity measure are introduced in Sections 2.1, 2.2 and 2.3, respectively.

2.1 Undersampling Methods

Undersampling techniques remove samples from the majority class while remain minority class unchanged in order to balance the dataset. Random undersampling techniques (RUS) randomly delete samples of the majority class until a balanced distribution is achieved. Compared with oversampling techniques, undersampling techniques use less time to train a classifier because they remove a large number of samples from the dataset, but also may lose important information. Results in [21] shows that the RUS performs well on imbalanced data with noise.

Some studies applied the RUS in the framework of ensemble models, e.g. RUSBoost[16], EasyEnsemble[17] and BalanceCascade[17]. The RUSBoost algorithm applies the RUS technique to generate several subsets of samples with the same number of minority and majority samples. Then each subset is used to train a weak classifier. Finally, all classifiers are combined as an ensemble system which votes to classify future unknown samples. The EasyEnsemble and the BalanceCascade are similar with the RUSBoost, but apply different strategies to generate subsets.

The EasyEnsemble is a method based on bagging algorithm, which divides the majority class into several subsets independently and trains classifiers with subsets combining the minority and the majority classes. On the other hand, the BalanceCascade works in a supervised way. It removes the correctly classified samples from the majority class in iteration such that the classifier would become more sensitive to the incorrectly classified samples in the next iteration. Another group of undersampling techniques based on the K-nearest neighbor classifier are proposed in [18], namely, NearMiss-1, NearMiss-2, NearMiss-3 and so on. They use the k-nearest neighbors to select samples from the majority class. The one-sided selection (OSS) method proposed in [19] is another undersampling algorithm which uses data cleaning technique to remove improper samples from the current training set.

2.2 Oversampling Methods

The simplest oversampling method is the random oversampling method (ROS) [1]. The ROS duplicates a certain number of minority class samples randomly and adds them to the dataset. This method is simple and effective in most cases, but likely to make overfitting. The SMOTE (Synthetic Minority Oversampling Technique) [3] generates synthetic minority instances by linear interpolation between adjacent minority samples. It generates samples in the same feature space of the original minority class which makes the re-sampled distribution more reasonable.

The SMOTE works as follows: Let T to be the set of all training samples and P to be the set of samples belonging to the minority class. For each minority sample x_i , one of its k-nearest neighbors is randomly selected and labeled as $x_{i,k}$. For each pair of x_i and $x_{i,k}$, a new sample is added between them which is generated as follows with a random number $\lambda \in (0, 1)$:

$$\mathbf{x}_{new} = \mathbf{x}_i + \lambda(\mathbf{x}_i - \mathbf{x}_{i,k}) \quad (1)$$

The SMOTE effectively improves the performance of classifiers on imbalanced datasets. However, it also has some disadvantages, e.g. linear interpolation is carried out only between minority samples and their k-nearest neighbors which will synthesize samples between majority and minority classes when the data is sparse and mixed to gather in the input space. Therefore, several methods are proposed to improve the SMOTE in different aspects.

The Borderline SMOTE (BLS)[4] deals with the samples located at the boundary of different classes only. In comparison to other methods, samples located near the boundary are more difficult to be correctly classified. So, it would be more effective to generate synthetic samples around them. The Borderline-SMOTE1 (BLS1) only creates new samples between the samples located at the boundary and their k nearest minority neighborhoods, while the Borderline-SMOTE2 (BLS2) also takes their majority nearest neighbors into account.

The Safe Level SMOTE (SLS) [5] assigns each positive sample with a value called safe level before generating synthetic samples. It generates synthetic samples closer to the minority samples with a larger safe level to make sure that they are locate in safe

regions. The safe level of a sample is computed by the number of minority samples in its k -nearest neighbors. If the safe level of a sample is close to 0, then it is likely a noise sample and can be discarded. If the safe level of a sample is close to k , then the instance is the safest and well located within the minority class. Hence, the safe level prefers creating new samples which is located well in the minority class and not being mixed with majority samples.

Although in most cases, the SLS can effectively covers the aforementioned shortage of the SMOTE, it may fail when the data distribution of the minority class is very sparse. In sparse dataset, very few minority samples will achieve a high safe level and therefore the SLS will perform poorly.

In contrast to the SLS, the Local Neighborhood SMOTE (LNS) [6] is more suitable for a sparse dataset. The SLS just randomly creates a sample from minority neighbors of samples in the minority class while the LNS takes also majority neighborhoods of samples in the minority class into account for generating new samples. As mentioned above, both the SLS and the LNS apply the concept of local neighbors and take the distribution within a local area around minority samples into account before generating synthetic samples.

2.3 The Sensitivity Measure and Q -Neighborhood

The Stochastic Sensitivity Oversampling (SSO) uses both the local neighbors and the stochastic sensitivity measure (ST-SM) to generate synthetic samples in a similar way to linear interpolation in the SMOTE.

In contrast to define a local neighborhood by the k -nearest neighbors as in both the SLS and the LNS, the SSO defines the local neighbor by a Q -neighborhood which is a hypercube centered at a minority sample with the radius of Q . The Q -neighborhood is originated from the Localized Generalization Error Model (L-GEM) [7] which is widely applied to active learning [8-11]. However, imbalanced datasets are not considered in [8-11] and therefore they may suffer from the same problems of all other machine learning techniques for imbalanced datasets. The Q -neighborhood ($S_Q(x_b)$) of the sample x_b is defined as follows:

$$S_Q(x_b) = \left\{ x \mid \begin{array}{l} x = x_b + \Delta x; \Delta x = (\Delta x_1, \dots, \Delta x_n)' \\ |\Delta x_i| \leq Q \quad \forall i = 1, \dots, n \end{array} \right\} \quad (2)$$

The union of $SQ(x_b)$ of all samples is denoted as the Q -Union. In [7], the stochastic sensitivity measure (ST-SM) is used to compose the L-GEM for architecture selection for classifiers. The ST-SM measures the expected squared output differences between training samples and unseen artificial samples located within the Q -Union. The ST-SM of a classifier is defined as follows:

$$E_{S_Q}((\Delta y)^2) = (1/N) \sum_{b=1}^N \int_{S_Q(x_b)} (\Delta y)^2 1/(2Q)^n dx \quad (3)$$

where $\Delta y = f_{\theta}(x) - f_{\theta}(x_b)$ and x denotes unknown samples located within the Q -neighborhood of x_b and N denotes the number of samples, respectively. The probability density function $1/(2Q)^n$ is used in the formula because all the unknown samples in the Q -neighborhood are assumed to have the same chance to appear, i.e. uniformly distributed.

In the SSO, the Q value is selected by the average distance between instances of the minority class and their k^{th} nearest neighbor. The SSO uses the hybrid value different metric to compute the value of Q , which uses a normalized Euclidean distance for continuous attributes and a value difference metric for discrete attributes [12].

3 The Proposed SSO

We firstly introduce basic concept of RBFNN and ST-SM in Section 3.1. Then, the SSO will be proposed in Section 3.2.

3.1 RBFNN and ST-SM

The SSO technique uses Radial Basis Function Neural Network (RBFNN) [13] as the classifier which consists of n input nodes, h hidden nodes, m output nodes. Let $x = (x_1, x_2, \dots, x_n)^T \in R^n$ be the input vector of the RBFNN, $W \in R^{h \times m}$ be the weight matrix between the hidden and output layers, $y = [y_1, \dots, y_m]^T$ be the output of the RBFNN, u_i and v_i be the center vector and the width parameter of the activation function of the i^{th} hidden neuron of the RBFNN. The Gaussian activation function is used:

$$\Phi_i(t) = e^{-\frac{t^2}{v_i^2}} \tag{4}$$

So the k^{th} output of the RBFNN is expressed as follows:

$$y_k = \sum_{i=1}^h w_i \phi_i(\|x - u_i\|) \tag{5}$$

The k -mean clustering method is used to determine the data centers (u_i) of the hidden layer. The width parameter is determined by the average of distances between nearest centers. Finally, we use the least square method to calculate the output weight vector.

The SSO uses the ST-SM as the indicator to measure the imbalance ratio in the local area around a sample of the minority class. When the sensitivity of a sample is large, there is a high possibility that the neighbors within its Q -neighborhood would be misclassified.

According to [14], we can obtain the sensitivity of the sample (x_b) by the following equation:

$$\begin{aligned}
 & E_{S_Q(x_b)}((\Delta y)^2) \\
 &= \int_{S_Q(x_b)} (f(x_b) - f(x))^2 p(x) dx \\
 &= \int_{S_Q(x_b)} (f^2(x_b) - 2f(x_b)f(x) + f^2(x)) p(x) dx \\
 &= f^2(x_b) - 2f(x_b) \int_{S_Q(x_b)} f(x) p(x) dx \\
 &\quad + \int_{S_Q(x_b)} f^2(x) p(x) dx \\
 &= f^2(x_b) - 2f(x_b)I_1 + I_2
 \end{aligned} \tag{6}$$

where $I_1 = \int_{S_Q(x_b)} f(x)p(x)dx$ and $I_2 = \int_{S_Q(x_b)} f^2(x)p(x)dx$

Assume that each feature of the input are independent to each other, we have:

$$I_1 = \frac{1}{(2Q)^n} \sum_{i=1}^h w_i \prod_{j=1}^n \left(\sqrt{\frac{\pi}{2}} v_i \left(\begin{array}{c} \operatorname{erf} \left(\frac{x_{bj} - u_{ji} + Q}{\sqrt{2} v_i} \right) \\ - \operatorname{erf} \left(\frac{x_{bj} - u_{ji} - Q}{\sqrt{2} v_i} \right) \end{array} \right) \right) \tag{7}$$

$$\begin{aligned}
 I_2 &= \int_{S_Q(x_b)} f^2(x) p(x) dx \\
 &= \frac{1}{(2Q)^n} \int_{S_Q(x_b)} \left(\sum_{i=1}^h w_i \exp \left(\frac{\sum_{j=1}^n (x_{bj} - u_{ji})^2}{-2v_i^2} \right) \right)^2 dx \\
 &= \left(\frac{\sqrt{\pi}}{4Q} \right)^n \sum_{i,r=1}^h \left(\begin{array}{c} \left(\frac{\sqrt{2v_i^2 v_r^2 (v_i^2 + v_r^2)}}{(v_i^2 + v_r^2)} \right)^n w_i w_r \exp \left(-\frac{1}{2} \frac{\sum_{j=1}^n (u_{rj} - u_{ji})^2}{v_i^2 + v_r^2} \right) \\ \prod_{j=1}^n \left(\begin{array}{c} \operatorname{erf} \left(\frac{(v_i^2 + v_r^2)(x_{bj} + Q) - (v_i^2 u_{rj} + v_r^2 u_{ji})}{\sqrt{2v_i^2 v_r^2 (v_i^2 + v_r^2)}} \right) \\ - \operatorname{erf} \left(\frac{(v_i^2 + v_r^2)(x_{bj} - Q) - (v_i^2 u_{rj} + v_r^2 u_{ji})}{\sqrt{2v_i^2 v_r^2 (v_i^2 + v_r^2)}} \right) \end{array} \right) \end{array} \right) \tag{8}
 \end{aligned}$$

3.2 Stochastic Sensitivity Oversampling Technique

The SSO computes the ST-SM for every minority sample. For a sample yielding a high sensitivity, more synthetic samples will be generated within its Q -neighborhood using the following Equation:

$$x_{new} = x_i + Q \times \lambda \tag{9}$$

Equation (9) makes sure that the synthetic samples are located around the minority samples. Iteratively, the SSO creates proper minority samples using the ST-SM calculated in the previous iteration and the oversampling ratio which are then added to the dataset. The procedures of the SSO are as follows:

<p>Input: Training dataset U ; Oversampling ratio θ ; Maximum number of iterations α .</p> <p>Output: a set of synthetic instances Z .</p>
<ol style="list-style-type: none"> 1. P = the set of minority instances; N = the set of majority instances; $Z = \Phi$; $t = 1$ m = the dimension of feature space 2. Do the loop: <ol style="list-style-type: none"> a) Calculate distances between minority samples and their k^{th} nearest neighbor, then calculate the radius (Q) of their Q-neighborhoods: $Q_i = \frac{1}{(n - \sqrt{m})} \sum_{i=1}^n x_{i,k} - x_i$ b) Train a RBFNN with dataset U , then calculate the ST-SM for each sample in P . c) Create $P \times \theta$ synthetic samples in P according to the STSMs of minority samples. Samples yielding larger STSMs will be selected to create synthetic samples in a higher chance. d) Generate synthetic samples according to $s_i = x_i + Q \times \lambda,$ where $x_i \in P$ and $\lambda \in (STSM - 1, 1 - STSM)$. Then add it to sets U and Z . e) $t = t + 1$ f) If $t > \alpha$, stop the iterations.

4 Experiments

In this section, the SSO is compared with five popular oversampling methods: random oversampling (ROS), the original version of the SMOTE (SMOTE), two versions of the borderline-SMOTE (BLS1 and BLS2), the Safe Level SMOTE (SLS) and the Local Neighborhood SMOTE (LNS). All methods are implemented with the *Matlab*.

The experiments are carried out on 6 datasets with different imbalance ratio and number of samples, which are selected from the UCI repository. For those datasets with several classes, we select one of them as the minority class while all the remaining classes are combined to serve as the majority class.

In addition to the average classification accuracy, we also use confusion matrix [15] to further analyze the classification performances of different classes, as illustrated in Figure 1. We label the majority class as the negative class and the minority class as the positive class. In the table of Figure 1, p, n, Y, N denote the true positive class label, the true negative class label, the predicted positive class and the predicted negative class, respectively.

	p	n
Y	TP (True Positives)	FP (False Positives)
N	FN (False Negatives)	TN (True Negatives)

Fig. 1. Confusion matrix

Precision, Recall, F-Measure and G-Mean are then computed based on the confusion matrix. They are defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

$$F - Measure = \frac{(1 + \beta)^2 \cdot recall \cdot precision}{\beta^2 \cdot recall + precision} \tag{12}$$

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{13}$$

The precision measures the ratio of correctly classified positive class samples over all samples being classified as positive. The recall measures the ratio of correctly classified positive class samples over all samples belonging to the positive class (true label). The F-Measure metric combines precision and recall as a single measurement to indicate the effectiveness of a classifier in terms of the ratio of the weighted importance on either recall or precision as determined by the user selected β coefficient. The G-mean evaluates the degree of inductive bias in terms of the ratio of accuracies of positive and negative classes.

Before experiments, an analysis of the datasets is performed. The analysis method proposed in the borderline SMOTE to label samples of the minority class as noise or dangerous, namely, borderline is used. We calculate m nearest neighbors of each sample x in the minority class, and m' is the number of majorities among them.

If $m/2 \leq m' < m$, then x is considered as a dangerous sample. If $m' = m$, then x is a noise sample. The characteristics of each dataset are listed in Table 1, where md and mn denote the dangerous samples and noise samples among m nearest neighbors, respectively. In our experiments, m is set to 5, which is a value usually used in related studies. Table 1 shows that all the datasets contain a number of dangerous samples, which can represent the most common situations in real life.

Table 1. Characteristic of the datasets

datasets	Number of attributes	number of samples	Number of minority	Imbalance Ratio	md	mn
ionos	34	176	63	0.35795	29	10
primary_tumor	23	173	42	0.24277	16	7
ionosphere	34	176	63	0.35795	28	11
german	61	500	150	0.3	106	19
horse	58	182	26	0.14286	15	10
pima	8	384	134	0.34896	83	13

Each dataset is randomly divided into training dataset and test dataset with roughly equal sizes for 30 times. Tables 2 and 3 show the average F-measure and the average G-mean over 30 times independent runs for different methods on different datasets. In the tables, bolded numbers in each row represents the best performing method. For SMOTE based methods, two parameters need to be selected: number of neighbors k and ratio of oversampling σ . In our experiments, $k = 5$ while σ is set to a multiple of 5, i.e. 5, 10, 15, 20 in turns. Among the four choices of σ , the one yielding the best result is selected for each method on each dataset. Tables 2 and 3 show that the SSO performs the best in 5 out of 6 datasets. For datasets with a number of dangerous and noise samples, the SSO generates synthetic samples around the minorities, while the ROS, the SMOTE, the BLS1 and the BLS2 generate new samples in the area of majorities which make a more screwed distribution. The SLS and the LNS are able to avoid this situation in some degrees, but the SSO perform better than them in our experiments.

Table 2. F-mean

datasets	ROS	SMOTE	BLIS1	BLS2	SLS	LNS	SSO
ionos	0.7572	0.7523	0.7509	0.7502	0.7540	0.7523	0.7610
primary_tumor	0.6557	0.6627	0.6571	0.6577	0.6594	0.6601	0.6759
ionosphere	0.7302	0.7151	0.7307	0.7326	0.7308	0.7269	0.7379
german	0.5686	0.5555	0.5358	0.5295	0.5550	0.5630	0.5808
horse	0.3640	0.3170	0.2069	0.2073	0.2396	0.2801	0.3699
pima	0.6535	0.6513	0.6659	0.6691	0.6597	0.6599	0.6445

Table 3. G-mean

datasets	ROS	SMOTE	BLS1	BLS2	SLS	LNS	SSO
ionos	0.7912	0.7884	0.7866	0.7860	0.7893	0.7874	0.8085
primary_tumor	0.7797	0.7576	0.7497	0.7492	0.7485	0.7515	0.7845
ionosphere	0.7703	0.7583	0.7724	0.7723	0.7775	0.7734	0.7919
german	0.6601	0.6710	0.6549	0.6503	0.6752	0.6788	0.6940
horse	0.6460	0.5126	0.3650	0.3649	0.3992	0.4495	0.6494
pima	0.6960	0.6984	0.7281	0.7317	0.7169	0.7114	0.7222

5 Conclusion

In this paper, we propose a new oversampling technique based on the concept of Q -neighborhood and the stochastic sensitivity measure of the Localized Generalization Error Model, i.e. the SSO. The SSO uses the stochastic sensitivity to select minority samples and create synthetic samples within the Q -neighborhoods centered at these samples. Such that, all synthetic samples are located close to other real minority samples. Experimental results show the effectiveness of the SSO.

Acknowledgments. This work is supported by a National Natural Science Foundation of China (61272201) and a Fundamental Research Funds for the Central Universities 10561201472 and a Student Research Project of South China University of Technology 105612014S468.

References

1. He, H., Garcia, E.: Learning from imbalanced data. *IEEE Transactions on Data and Knowledge Engineering* **21**(9), 1263–1284 (2009)
2. Ng, W.W.Y., He, Z.-M., Yeung, D.S., Chan, P.P.K.: Steganalysis Classifier Training via Minimizing Sensitivity for Different Imaging Source. *Information Science*, 211–224 (2014)
3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. *J. of Artificial Intelligence Research* **16**, 341–378 (2002)
4. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005*. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
5. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In: Theeramunkong, T., Kijssirikul, B., Cercone, N., Ho, T.-B. (eds.) *PAKDD 2009*. LNCS, vol. 5476, pp. 475–482. Springer, Heidelberg (2009)
6. Maciejewski, T., Stefanowski, J.: Local neighbourhood extension of smote for mining imbalanced data. In: *Proceedings of 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 104–111 (2011)

7. Yeung, D.S., Ng, W.W.Y., Wang, D., Tsang, E.C.C., Wang, X.-Z.: Localized Generalization Error and Its Application to Architecture Selection for Radial Basis Function Neural Network. *IEEE Trans. on Neural Networks* **18**, 1294–1305 (2007)
8. Tejchman, J., Kozicki, J.: General. In: Tejchman, J., Kozicki, J. (eds.) *Experimental and Theoretical Investigations of Steel-Fibrous Concrete*. SSGG, vol. 3, pp. 3–26. Springer, Heidelberg (2010)
9. Chan, P.P.K., Ng, W.W.Y., Yeung, D.S.: Active Learning using Localized Generalization Error of Candidate Sample as Criterion. In: *IEEE Proceedings of International Conference on Systems, Man and Cybernetics*, pp. 3604–3609 (2005)
10. Yeung, D.S., Zhang, Y., Ng, W.W.Y., Chen, Q.-C.: Active Learning using Localized Generalization Error for Text Categorization. In: *Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 2686–2691 (2006)
11. Ng, W.W.Y., Yeung, D.S., Cloete, I.: Input Sample Selection for RBF Neural Network Classification Problems using Sensitivity Measure In: *IEEE Proceedings of International Conference on Systems, Man and Cybernetics*, pp. 2593–2598 (2003)
12. Wilson D.R., Martinez, TR.: Improved heterogeneous distance functions. arXiv preprint cs/9701101 (1997)
13. Buhmann, M.D.: *Radial Basis Functions: Theory and Implementations*. Cambridge University (2003)
14. Sun, B., Ng, W.W.Y., Yeung, D.S., Wang, J.: Localized Generalization Error based Active Learning for Image Annotation. In: *IEEE Proceedings of International Conference on Systems, Man and Cybernetics*, pp. 60–65 (2008)
15. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Data Mining Researchers Technical Report HPL-2003-4, HP Labs (2003)
16. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., et al.: RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **40**(1), 185–197 (2010)
17. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**(2), 539–550 (2009)
18. Mani, I, Zhang, I.: kNN approach to unbalanced data distributions: a case study involving information extraction. In: *Proceedings of Workshop on Learning from Imbalanced Datasets* (2003)
19. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-sided Selection//*ICML*. **97**, 179–186 (1997)
20. Van Hulse, J., Khoshgoftaar, T.: Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering* **68**(12), 1513–1542 (2009)

Application to Detection

A Heterogeneous Graph Model for Social Opinion Detection

Xiangwen Liao^{1,3}, Yichao Huang^{1,3}, Jingjing Wei^{2,3},
Zhiyong Yu^{1,3}, and Guolong Chen^{1,3(✉)}

¹ College of Mathematics and Computer Science, Fuzhou University,
Fuzhou 350116, China

liaoqxw@fzu.edu.cn, fjnphyc@qq.com, yuzhiy@gmail.com

² College of Physics and Information Engineering, Fuzhou University,
Fuzhou 350116, China

weijj_0517@163.com, cgl@fzu.edu.cn

³ Fujian Provincial Key Laboratory of Network Computing and Intelligent
Information Processing, Fuzhou University, Fuzhou 350116, China

Abstract. Microblogging services, such as Twitter, have become popular for people to share their opinions towards a broad range of topics. It is a great challenge to get an overview of some important topics by reading all tweets every day. Previous researches such as opinion detection and opinion summarization have been studied for this problem. However, these works mainly focus on the content of text without taking the quality of short text and features of social media into consideration. In this paper, we propose a heterogeneous graph model for users' opinion detection on microblog. We first extract keywords of topics. Then, a three-level microblog graph is constructed by combining user influence, word importance, post significance, and topic periodicity. Microblog posts are ranked from different topics by using the random walk algorithm. Experimental results on real a dataset validate the effectiveness of our approach. In comparison with baseline approaches, the proposed method achieves 8% improvement.

Keywords: Random walk · Opinion detection · Social media · Microblog graph

1 Introduction

Recently, microblog has become one of the most popular social networking sites. For example, Twitter¹ has rapidly gained worldwide popularity with over 200 million active users monthly who generate over 500 million tweets daily in 2013². It enables people to freely post short messages up to 140 characters. Weibo in China has similar phenomenon to Twitter. People share their daily information, keep in touch with their friends, and exchange opinions towards a broad range of topics on microblog. The large amount of microblog posts makes information acquisition difficult. For a user, it is an

¹ <http://twitter.com>

² <http://blog.twitter.com/2013/>

almost impossible task to find the major opinion of a topic on microblog sites by reading all contents every day. Moreover, many posts on microblogs are not very user-friendly to read or analyze because of their informal writing styles. It costs a lot of time to collect useful information about a topic. Consequently, a service is urgently needed to select meaningful and representative ones from the tremendous volume of posts. This service can be used for market researches, forecasting votes, advertising analysis and network content security, etc.

Many universities and research institutes have shown great interests in mining opinions from a large-scale user data. Since 2006, NIST and DARPA in USA have sponsored and organized the TREC (Text Retrieval Conference) including Blog Opinion Retrieval. Starting from the same time, JSPS and NII in Japan have sponsored the NTCIR (NII Test Collection for IR Systems), which has paid attention to opinion mining. ICT, Shanghai Jiaotong University, Fuzhou University, and other universities have also continuously organized COAE for Chinese opinion mining in China since 2008.

In this paper, we focus on opinion detection and opinion summarization, which are to find the most significant posts with opinions to a given topic. Current works usually tap views from the Internet population according to the text content on a social media. However, text on microblog is short, informal and posted freely by users. Additionally, users and information are two important dimensions of social media. Both of them cannot be ignored. Graph-based approach such as phrase graph [17] and LexRank [18] describe the salience of the short text but are not the best features on social media. Therefore, we propose a heterogeneous graph model for users' opinion detection on microblog. We first extract keywords of topics. Then, a three-level microblog graph is constructed by combining user influence, word importance, post significance, and topic periodicity. Afterwards, we rank the microblog posts from different topics by using the random walk algorithm. Finally, we select the top 10 microblog posts of each topic as the users' main idea for this topic. Experiments in this paper are conducted on a real dataset containing thousands of posts. In comparison with baseline approaches, experimental results show that our proposed approach achieves better performance.

The rest of this paper is organized as follows. Related work is introduced in Section 2. Detailed introduction of our graph-based model for opinion detection is given in Section 3. Section 4 shows experiments and results. Finally, we conclude this work in section 5.

2 Related Work

2.1 Content-Based Opinion Mining

Opinion mining has been widely studied in industry and academia, such as opinion recognition [1], opinion retrieval [2], opinion elements extraction [3], and comment spam recognition [4], etc. Previous work about opinion mining can be categorized into two types: statistical model and retrieval model. The former made full use of documents' information, such as opinion lexicon [5], contextual information [6], sentence-based information [7], proximity of words [8], background knowledge of words

[9] and so on, while the latter used unified retrieval algorithms to retrieve the opinions of users on specific topics. For typical examples, there is lexicon-based sentiment for opinion retrieval [10], opinion words expansion for relevance model [11], topic model combined with topics and opinions [12], etc. Benefit from the solid theoretical foundation, opinion mining based on unified retrieval model is easy to understand.

Nevertheless, these current researches mainly analyzed users' opinions from the contents of text without the features of social media. In fact, the contents of text in social media are usually short and informal, which makes it difficult for opinion mining. Fortunately, some features like the relationship of reposting on microblog posts and the relationship of supporting on users can be quite useful for mining results improvement.

2.2 Graph-Based Approach

In order to incorporate features of social media, Graph-based approaches are proposed to model the relationship between objects or items and ranking them. These approaches built the graph of web links to rank the web pages, such as PageRank [13], HITS [14], LexRank [15], etc.

With respect to microblog, the opinion summarization of microblog is the most similar research to our work. The phrase graph algorithm is the most frequently studied graph-based approach in microblog opinion summarization. Beaux Sharifi, Hutton and Kalita [16] proposed a phrase reinforcement summarization algorithm on leverage of trending phrase and phrases specified by a user in microblog posts. It achieves substantial improvements on ROUGE results by taking advantage of the link structure among words. Nichols et al. [17] generate journalistic summary for events in world cup games by applying phrase graph algorithm only to the longest sentence in each tweet. Additionally, PageRank-like algorithms such as LexRank and TextRank have also been investigated by Inouye and Kalita [18].

However, these graph-based works mainly pay attention to construct graph with the features of the text. The nodes in the graph were denoted by sentences or phrases, which cannot take advantages of user influence and topic periodicity.

3 Microblog Opinion Detection with Heterogeneous Graph Model

The problem of opinion detection on microblog is formulated as follows: given a social media S and a topic list, we can obtain a microblog post set $P = \{p_1, p_2, \dots, p_N\}$ containing the topic, where N is the number of the posts. Each post is attached with a time mark and the user who publish the post, then we can obtain a user set $U = \{u_1, u_2, \dots, u_K\}$ containing the user information; where K is the number of user. We use a graph-based model to mine on P and find the main opinions. Specifically, the main opinions are represented by the most M representative posts for each topic.

To generate main opinions for a topic, there are three steps: First, a statistical method similar to the TF*IDF is used to detect the keywords of the topics. Second, we define the measure methods for scoring the microblog posts, users, and words. After that, we construct a three-level graph to rank the posts for each topic by using random walks algorithm.

3.1 Topic Keywords Detection

According to different topics, there are different keywords for them. For example, the players’ name may be the keywords of NBA; “flowers” and “love” may be the keywords of Mother’s Day. We observe that these keywords have different statistics among the different topics. Usually, term frequency of the keywords put up significant differences to different topics. Therefore, we take the posts for a same topic as a document and calculate the TF*IDF:

$$tf_j(w_i) = \log n_{i,j} \tag{1}$$

$$idf(w_i) = \frac{|T|}{|\{j : w_i \in t_j\}|} \tag{2}$$

We redefine the formula because of two reasons:

- We regard all posts in a topic as a document. The phenomenon of word sparse becomes serious. Therefore we remove the denominator and calculate the logarithm about the “*tf*”.
- The number of the topics is much smaller than documents. Hence, we cannot calculate the logarithm about the “*idf*”.

Variable *t* expresses a sub-set of P separated by topic. $tf_j(w_i)$ is the term frequency of the i^{th} word at the j^{th} topic. *T* is the number of topics.

$tf_j(w_i) * idf(w_i)$ is used to describe the importance of the i^{th} word to post set of j^{th} topic, then a threshold is set to detect the keywords for each topic. Parts of the statistical results are below:

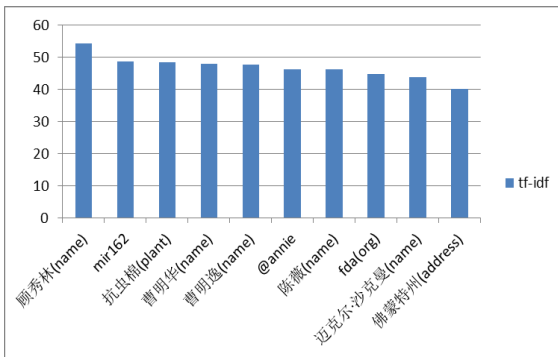


Fig. 1. Top 10 keywords for a topic

According to the Pareto principle, we select about the top 20 percent of the words as the topic keywords.

3.2 Heterogeneous Graph Model for Opinion Detection

A post is salient if it is posted by influential users, similar to most of the posts in the same topic, contains rich useful information and presented in a good writing style.

The first is emphasized because microblog is a social network, in which influential users have more followers and more likely to dominate evolution of the topic. A post which is similar to most of the posts in the same topic often expresses group’s view and probably shows the influential view, so we present the second. The last is set because the salient post is more valuable and readable. Then the salientest posts will be selected to construct the mining result.

In this paper, a three-level graph model is proposed to rank the posts by taking advantage of relations among words, posts, and users. Figure 2 shows the overview of the proposed model.

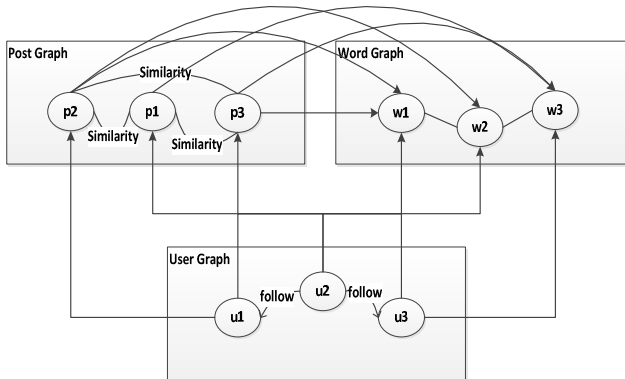


Fig. 2. The Three-level graph based model

The model is formed with three parts in a unified way for microblog post, word, and user respectively. When ranking one of them, the result will be affected by the other two. In the model, microblog posts are connected with each other through syntactic similarities. If the cosine similarity of two microblog posts is not equal to zero, there is a link between them. If two words appeared in a same post, a link between them is created. And users are associated by the following-follower relationship, if user u_i follows user u_j , there is a link from u_i to u_j . Besides, there are links among three levels: if p_i is posted by u_j , we connect u_j with p_i ; if the w_i appears in p_j , p_j is linked to w_i ; Similarly, if w_i is in the posts published by u_j , u_j is linked to w_i .

According to the definition, the ranking score of one post in a topic is defined as:

$$\begin{aligned}
 Score^{(r+1)}(p_i) &= \alpha_1 \sum_{p_j \in adj[p_i]} \frac{Sim(p_i, p_j)}{\sum_{p \in adj[p_i]} Sim(p_j, p)} \\
 &+ \beta_1 \cdot \sum_{w \in p_i} Score^{(r)}(w) + \gamma_1 \cdot Score^{(r)}(u_i)
 \end{aligned} \tag{3}$$

where $Score^{(r)}(w)$, $Score^{(r)}(p)$, and $Score^{(r)}(u)$ represent the ranking score of word w microblog post p and user u in r^{th} iteration respectively. In this paper, p is represented in two forms:

$$\begin{aligned}
 p &= \langle w_1, w_2, \dots, w_n \rangle \\
 p' &= \langle w_1, w_2, \dots, w_n, t \rangle
 \end{aligned}$$

where w_i denotes the keyword of a topic, t is a time stamp. With consideration of the topic periodicity, second form is defined.

$Sim(p_i, p_j)$ is the cosine similarity between p_i and p_j , $adj[p_i]$ denotes the microblog posts linked to p_i directly, w is the words in p_i and u_i is the author of p_i .

A user is influential if he connects to other influential users, uses important words to post the typical posts. So the ranking score of a user is defined as,

$$\begin{aligned}
 Score^{(r+1)}(u_i) &= \\
 &\alpha_2 \sum_{p \in P_{u_i}} Score^{(r)}(p) + \beta_2 \cdot \sum_{w \in p, p \in P_{u_i}} Score^{(r)}(w) \\
 &+ \gamma_2 \cdot \sum_{u_j \in flw[u_i]} \frac{1}{|frd[u_j]|} Score^{(r)}(u_j)
 \end{aligned} \tag{4}$$

where P_{u_i} denotes the posts published by u_i , w are the words of the posts published by u_i . $flw[u_i]$ represents the followers of u_i and $frd[u_j]$ refers to the users u_j follows.

A word is important if it connects to other important words, contained in the typical posts, used by influential users,

$$\begin{aligned}
 Score^{(r+1)}(w_i) &= \alpha_3 \sum_{p \in P_{w_i}} Score^{(r)}(p) \\
 &+ \beta_3 \cdot [(1 - d) \cdot \frac{tf \cdot idf(w_i)}{\sum_{w_j \in p} tf \cdot idf(w_j)} \\
 &+ d \cdot \sum_{w_j \in adj[w_i]} \frac{1}{|adj[w_j]|} Score^{(r)}(w_j)] \\
 &+ \gamma_3 \cdot \sum_{u \in U_{w_i}} Score^{(r)}(u)
 \end{aligned} \tag{5}$$

where P_{w_i} represents the posts containing w_i , w_j is the word occurs with w_i in the same post and $tf \cdot idf(w_i)$ denotes the value we have calculated in section 3.1; $adjl w_j J$ represents the words connecting to w_i and U_{w_i} refers to users who used w_i . Here, d is the damping factor which is set to 0.85 as described in PageRank algorithm [13]. The model is inspired by Wei et al.’s work [19] and we use the α_i , β_i and γ_i as the initial value refers to their work.

4 Experiment

4.1 Data Set

In the experiment, we use the real data set from Weibo, which is the most popular microblogging service in china. In total, we obtain 84113 posts about 12 topics published between February 2011 and May 2014 using the Weibo API. With consideration of social influence, we also obtain users’ information, the following-followed relationship and Weibo repost relationship.

The basic statistics of the data set including 12 topics is shown in Table 1.

Table 1. Basic statistics of data set

Topic Id	Weibo	Topic Id	Weibo
1	7572	7	6364
2	10886	8	6835
3	11569	9	5625
4	8080	10	5955
5	8935	11	7515
6	1514	12	3263

Generally, there are many noisy posts in the real data set, we filter the posts beforehand if they satisfy one of the following conditions:

- Including the link “http://t.cn”.
- Less than three words.
- Including the Chinese Onomatopoeia.
- Including the special Chinese word used for advertisement.

In order to label the microblog posts, we have ranked the posts by the topic relevance based on keywords of the topic. However, labeling top 10 typical posts from thousands of posts is too difficult for a volunteer to do. In order to solve this problem, every 10 posts are regarded as a group and people are asked to label the most salience post of groups. Similar to other topics, one of the topics is chosen to do the experiment.

4.2 Evaluation of the Graph-Based Approach

We use ROUGE as evaluation metric because the baseline LexRank used it and it is the important metric for opinion summarization. ROUGE-n is defined as:

$$ROUGE = \frac{\sum_{p \in S} \sum_{gram_n \in p} Count_{match}(gram_n)}{\sum_{p \in S} \sum_{gram_n \in p} Count(gram_n)} \quad (6)$$

where S is the reference set, n is the word length of n-gram. $Count_{match}(gram_n)$ is the maximum number of n-grams appearing both in the result generated by our framework and the reference set.

In order to demonstrate the effectiveness of our proposed model, we compared it with the following models:

- (1) LexRank[18]: A PageRank-like summarization algorithm, which calculates sentence salience using the random walk model.
- (2) MRGM: Our proposed mutual reinforcement graph model, which calculates user influence, word importance, and post significance.
- (3) MRGM+TP: On the basis of MRGM, take topic periodicity into the computation of the post significance.

First, we illustrate how the damping factor d influences the performance of ROUGE-1 in our work. The result is given in Figure 3.

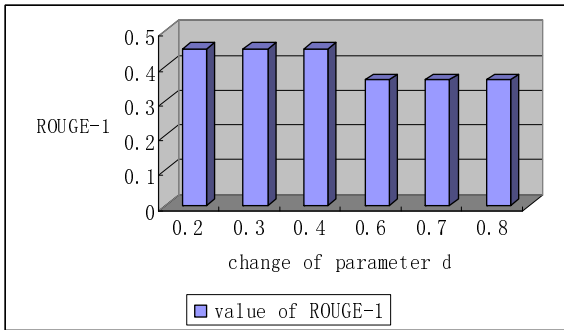


Fig. 3. Performance of ROUGE with varying d

We can find that the performance of ROUGE-1 achieve a higher value when d is less than 0.4. Thus, in the following experiments we set $d = 0.3$.

Next, the parameters α_i , β_i and γ_i are used to balance the relative weight of posts, words, and users. In our model, we set these parameters multiplied by several same ratios and try to find the influences of them. The result is given in Figure 4.

We can find that MRGM performs better when parameters are divided by 1.2 and without using topic periodicity. On the other hand, when the parameters are divided by 1.5, there is a higher value for MRGM+TP. Hence, in the following experiments, parameters are set according to these results.

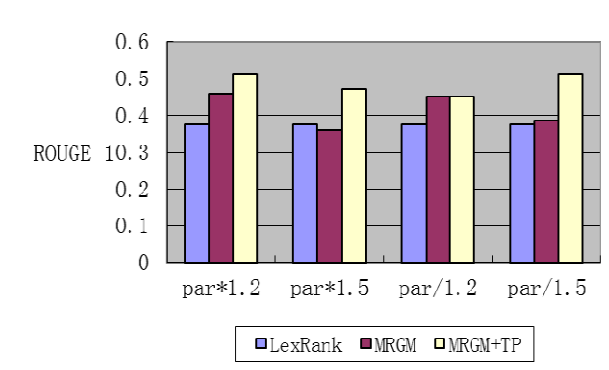


Fig. 4. Performance of ROUGE with varying parameters

Table 2 shows that our proposed model MRGM achieves higher ROUGE-1 value than baseline approach. It indicates user influence and word importance can help improve the performance of opinion detection. The result of MRGM+TP outperforms MRGM. It shows that the posts for topics have periodic features that similar opinions are always published in a same period.

Table 2. Experimental results

Approach	<i>ROUGE-1</i>
LexRank	0.375
MRGM	0.4578
MRGM+TP	0.5106

5 Conclusions

In this paper, we have proposed a heterogeneous graph model, which detects main opinion on Microblog. We decompose the microblog into three levels: word, post, and user. Firstly, we filter topic-related posts from the whole data set and extract keywords of topics. Then, we construct a three-level microblog graph, which combines user influence, word importance, post significance, and topic periodicity. Eventually, we rank posts by using the random walk algorithm. Experimental results show that the top posts can express the mainstream opinions of the topic. In the future, we will further study the topic evolution on more massive datasets and try to use MapReduce to make our approach scalable to big data environment.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China (No. 61300105), the Research Fund for Doctoral Program of Higher Education of China (No. 2012351410010), the Key Project of Science and Technology of Fujian (No. 2013H6012), the Project of Science and Technology of Fuzhou (No. 2012-G-113, 2013-PT-45), and the Scientific Research Project of the Educational Department in Fujian Province (No. JA10055).

References

1. Zhu, J., Wang, H., Zhu, M., et al.: Aspect-Based Opinion Polling from Customer Reviews. *IEEE Transactions on Affective Computing* **2**(1), 37–49 (2011)
2. Gerani, S., Carman, M.J., Crestani, F.: Proximity-Based Opinion Retrieval. In: *Proceeding of SIGIR2010 Conference*, Geneva, Switzerland, pp. 403–410 (2010)
3. Zhai, Z., Liu, B., Zhang, L., et al.: Identifying Evaluative Opinions in Online Discussions. In: *Proceedings of AAAI 2011 Conference*, San Francisco, California, USA, pp. 3434–3439 (August 2011)
4. Wang, G., Xie, S., Liu, B., et al.: Review Graph Based Online Store Review Spammer Detection. In: *Proceedings of IDCM 2011 Conference*, Vancouver, BC, Canada, pp. 1242–1247 (2011)
5. Qiu, G., Liu, B., Bu, J., et al.: Expanding Domain Sentiment Lexicon through Double Propagation. In: *Proceedings of IJCAI 2009 Conference*, California, USA, pp. 1199–1204 (July 2009)
6. Li, B., Zhou, L., Feng, S., et al.: A Unified Graph Model for Sentence-Based Opinion Retrieval. In: *Proceedings of ACL2010 Conference*, Uppsala, Sweden, pp. 1367–1375 (July 2010)
7. Wu, Y., Zhang, Q., Huang, X., et al.: Structural Opinion Mining for Graph-based Sentiment Representation. In: *Proceedings of EMNLP2011 Conference*, Edinburgh, UK, pp. 1332–1341 (July 2011)
8. Santos, R.L., He, B., Macdonald, C., Ounis, I.: Integrating Proximity to Subjective Sentences for Blog Opinion Retrieval. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) *ECIR 2009*. LNCS, vol. 5478, pp. 325–336. Springer, Heidelberg (2009)
9. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In: *Proceedings of SIGKDD 2009 Conference*, Paris, France, pp. 1275–1284 (June 2009)
10. Zhang, M., Ye, X.: A Generation Model to Unify Topic Relevance and Lexicon-based Sentiment for Opinion Retrieval. In: *Proceedings of SIGIR2008 Conference*, Singapore, pp. 412–418 (July 2008)
11. Huang, X., Croft, W.B.: A Unified Relevance Model for Opinion Retrieval. In: *Proceedings of CIKM 2009 Conference*, Hong Kong, China, pp. 947–956 (November 2009)
12. Mei, Q., Xu, L., Wondra, M., et al.: Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In: *Proceedings of WWW2007 Conference*, Banff, Alberta, Canada, pp. 171–180 (May 2007)
13. Page, L., Brin, S., et al.: The PageRank citation ranking: Bringing order to the web Technical report, Stanford Digital Library Technologies Project (1998)
14. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* **46**(5), 604–632 (1999)
15. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* **22**(1), 457–479 (2004)

16. Sharifi, B., Hutton, M.-A., Kalita, J.: Summarizing Microblogs Automatically. In: Proceedings of HLT2010 Conference, Los Angeles, California, USA, pp. 685–688 (June 2010)
17. Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using twitter In: Proceedings of IUI2012 Conference, Lisbon, Portugal, pp. 189–198 (February 2012)
18. Inouye, D., Kalita, J.K.: Comparing Twitter Summarization Algorithms for Multiple Post Summaries. In: Proceedings of SocialCom/PASSAT2011 Conference, Boston, MA, USA, pp. 298–306 (October 2011)
19. Wei, F., Li, W., Lu, Q., He, Y.: Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization In: Proceedings of SIGIR 2008 Conference, New York, NY, USA, pp. 283–290 (2008)

A Storm-Based Real-Time Micro-Blogging Burst Event Detection System

Yiding Wang, Ruifeng Xu^(✉), Bin Liu, Lin Gui, and Bin Tang

Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School,
Harbin Institute of Technology, Shenzhen 518055, China
weden1226@gmail.com, xuruifeng@hitsz.edu.cn

Abstract. Micro-blogging is becoming an important information source of breaking news event. Since micro-blogs are real-time unbounded stream with complex relationships, traditional burst event detection techniques do not work well. This paper presents the RBEDS which is a real-time burst event detection system following Storm distributed streaming processing framework. K-Means clustering approach and burst feature detection approach are performed to identify candidate burst events, respectively. Their outputs are incorporated to generate final event detection results. Such operation is implemented as a Storm Topology. The proposed system is evaluated on a large Sina micro-blogging dataset. The achieved system performance shows that the RBEDS system may detect burst events with good timeliness, effectiveness and scalability.

Keywords: Burst event detection · Distributed stream processing · Storm

1 Introduction

As a new social media services, micro-blogging is becoming the most popular social applications in the world. For example, Sina micro-blogging is the most popular micro-blogging site in China. By the end of April 2014, Sina has more than 500 million registered users while the number of active user is higher than 110 million. People share their information within 140 characters anytime and anywhere. Different from other social applications, micro-blogging has shown strong media attributes. Naturally, micro-blogging is becoming the new sources of information, especially for the burst news events.

Compared with traditional media services, micro-Blogging has shown to have the following features:

1). *Large scale.* Take Sina micro-blogging for example, the average number of new generated micro-blogs is higher than one hundred million in one day.

2). *Unbounded stream.* Micro-blogs are generated continuously by millions of users. Thus, it is impossible to predict the coming of next micro-blog.

3). *Real-time.* With the popularity of mobile computing devices, people can tweet anytime and anywhere. Its leads to the burst events to be reported in real-time.

4). *Social relationship.* Micro-blogging users are connected by large and complex social relations (such as friend and forwarder) and operation relations (such as forwarding and recommending).

Traditional event detection approaches, such as the methods proposed in Topic Detection and Tracking (TDT) evaluation for news text [1],[2],[3], are difficult to handle the micro-blogging data stream. Especially, the abundant social attributes such as poster's VIP information and comment status, and operation attributes such as forwarding, are seldom used in traditional event detection. In addition, the large-scale real-time micro-blogging steam always makes the traditional event detection approaches inefficient which leads to the burst event cannot be detected timely.

Storm is a real-time distributed computation framework for processing large amounts of input data with high fault-tolerance and horizontal scaling. Stream is the key abstract of Storm which consists of an unbounded sequence of Tuples. These Tuples can be created and processed in a distributed way. Tuples emission is the message passing mechanism in Storm which contains a sequence of values of any type. In Storm, each application is pre-defined as a network of Spouts and Bolts before it runs. Such a network is called a Topology which contains the whole application logic. It is helpful to process steam micro-blogging in real-time.

In this paper, we presents the RBEDS system, a real-time burst event detection system following Storm distributed streaming processing framework. In this system, the K-Means clustering based burst event detection and burst feature detection are performed on Storm, respectively. The outputs of Storm are incorporated to generate final burst events results. In addition, considering the data sparseness, dimensionality reduction based on locality sensitive hashing (LSH) is applied in the stage of K-Means clustering. Such framework is helpful to combine the advantages of Storm and event detection approaches. The proposed system is evaluated on a large micro-blogging dataset, which contains about 3.5 million micro-blogs, from three aspects, namely timeliness evaluation, effectiveness evaluation and scalability evaluation. Experiment results show that RBEDS system detects burst event effectively with good performance.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 presents the framework of RBEDS system. The Real-time distributed burst event detecting approach as a Topology in Strom is described in Section 4. Section 5 gives the evaluation and finally, Section 6 concludes.

2 Related Works

2.1 Burst Event Detection

There are many researches on burst event detection from news text while the detection from micro-blogging is still a relatively new research field. Generally, burst event detection techniques are camped into text-centric approaches [4],[5] and burst feature-centric approaches [6],[7],[8],[9].

Normally, text-centric approach firstly clustering the text and then extracts the burst features in the cluster. Using these burst features, the burst events are detected. This approach performs well on news text, but not very well on micro-blogging because micro-blogging text contains much spam messages which leads to a lot of noises in clustering. In addition, micro-blogs contain less content which leads to serious data sparseness.

Burst feature-centric approach firstly extracts and groups the burst characteristics and features. The burst feature set is then applied to recognize burst events. This approach may avoid the data sparseness, but cannot handle the spam problem.

2.2 Distributed Stream Processing for Large-Scale Text

Nowadays, researches on big data batch computing are relative mature. The typical researches are MapReduce [11] programming model and open source Hadoop computing system.

Hadoop is a typical big data batch computing framework. HDFS distributed file system is responsible for the storage of static data [10]. The calculation logic is assigned to each data nodes, via MapReduce, for data computing. This kind of “store-then-process” computing model has the shortcoming of time delay [12],[13] which affects the real-time burst event detection. Moreover, this framework cannot store all of the useful data since it cannot determine when the new data comes.

A new kind of big data processing model, namely large-scale data stream processing model, is investigated. Target to the characteristics of large-scale stream data, such as real-time, non-volatile, burstness and randomness, the new distributed stream processing platforms were proposed including Storm and S4 [14]. Corresponding to MapReduce running jobs [15], Storm runs topologies. One major difference between them is that a MapReduce job eventually finishes, whereas a Storm topology needs explicit termination to stop running.

3 System Overview and Architecture

Considering that the micro-blog data is unbounded stream data, a new burst event detection system following Storm framework, named RBEDS system, is designed. Its architecture is shown in Figure 1. The RBEDS system process micro-blog data stream in the following three steps:

Step 1. Data pre-processing. In this step, spam filtering and Chinese words segmentation/POS tagging are conducted. The spam filtering aims to filter out the advertisement and other noises that are harmful to burst event detection. Here, a binary classifier is developed to classify whether a micro-blogging belongs to noise data. The advertisement-related word features are extracted from 2,000 manually identified advertisement micro-blogging for the classifier. To avoid filtering out important micro-blogs, the classification model focuses on high precision.

Step 2. Burst event detection. In this step, the burst words detection and text clustering are conducted, respectively. Different from traditional text-centric burst event detection approach which does either text clustering or exact burst words, the outputs of burst words detection and text clustering are incorporated to generate the final burst event detection results. This is the main component of RBEDS. The details of Step2 will be given in Section 4.

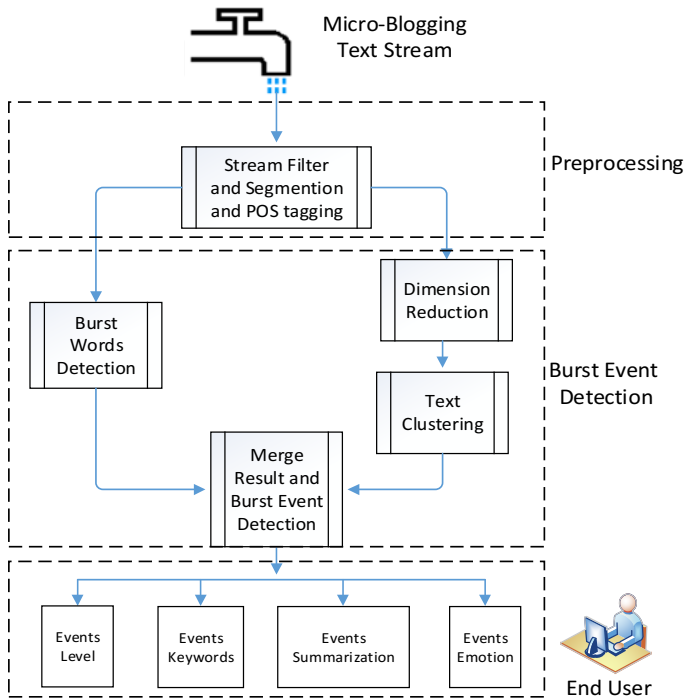


Fig. 1. Architecture of RBEDS

Step 3. Results visualization. “Event” is a specific concept. When we detected a new event happened, the end users often want to know its properties such as event level (e.g. major, secondary, general), abstract description, key words and event polarity judgment (e.g. positive or negative). Therefore, in this step, we extract the characteristics for the detected events, such as event level, abstract, key words and emotional tendency etc. They are visualized to the end user.

4 Burst Event Detection as a Storm Topology

This section describe the details of burst event detection in RBEDS system including dimension reduction, text clustering, burst words detection, result merge and burst event detection.

4.1 Dimension Reduction and Text Clustering

As shown in figure 2, there are three phases in this sub-step:

- 1). Dimensions reduction and vectorization phase;
- 2). Cosine distance calculation phase;
- 3). K-Means clustering phase.

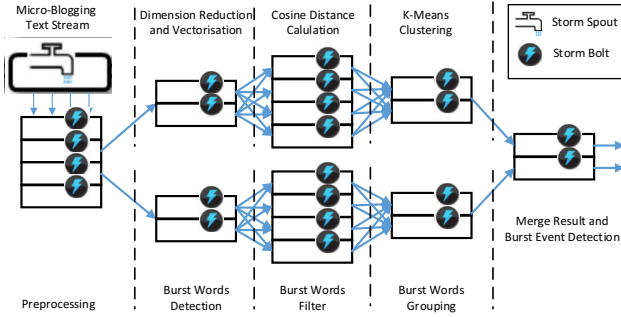


Fig. 2. Storm topology for burst event detection

In the second phrase, cosine distance is calculated to measure the similarity between different micro-blogs in two Bolts. The cosine similarity between two vectors u and v is defined as:

$$\cos(\theta) = \frac{\sum_{k=1}^n u_i \times v_i}{\sqrt{\sum_{k=1}^n (u_i)^2} \times \sqrt{\sum_{k=1}^n (v_i)^2}} \quad (1)$$

Firstly, we use one Bolt to calculate all the micro-blog's local cosine distance in each bucket. Secondly, we identify the similar micro-blog collection. The cosine distances between these similar micro-blogs are output to the next processing unit. As shown in figure 2, this Bolt was used to combine previous one. The most similar micro-blog collections in the update micro-blogging stream are then extracted and finally sent to the next K-Means clustering phrase.

In the third phrase, there are two Bolts. Firstly, we assign the parameter k in K-Means algorithm to represent the numbers of clusters. When the Tuple is received from the former stage, this Bolt determines whether adding the new incoming document into an existing cluster according to the threshold i . If the cluster already exists and its size is larger than m , the new document is appended into this cluster and then sent to the integration phase (See section 4.4). If the cluster exists for a long time and no new documents are added in a time-window (the time threshold is empirically set for one hour), this cluster is removed while the corresponding event is regarded terminated. When an eligible cluster is sent to the next phase, it will be existed in this stage until it is removed since no new micro-blogs are added. The next Bolt is a key word extraction module which is applied to extract the key words (up to 10 words) from every cluster. They are sent to burst event detection phrase as features provided by text clustering approach.

4.2 Burst Words Detection

To improve the accuracy of K-Means clustering algorithm, in this sub-step, the burst features are extracted. This sub-step includes three phases while each phase contains a Bolt.

In general, when a burst event occurs, its relevant vocabulary will become more active than usual. The burst words are detected based on the growth rate of some special words. Here, we only detect the noun and verb words which have real meaning in this part to avoid the influence from meaningless words. Firstly, the rate of related nouns and verbs are calculated with the consideration of micro-blogging forwarding numbers. Here, the first Bolt will keep the growth rate of every word in t hours (one hour) and then the results of this Tuples are sent to the next Bolt to consolidate and identify the burst words. p burst words are kept in this Bolt. The results are sent to the next phase for grouping the burst words. In the second phase, incremental single-pass clustering algorithm is applied to cluster burst words. The distances between each word and each cluster are set to the average distances of every word in the cluster. The results of clustering as burst words group are sent to integration phase.

4.3 Results Merge and Burst Event Detection

In this sub-step, the results of the above two sub-steps are merged, the results of burst event detection are then sent to the visualization step. This sub-step integrates the two results by computing the degree of overlapping. If the overlapping rate is higher than a threshold, the cluster is identified as a burst event. The corresponding micro-blog cluster is sent to results visualization step. In order to obtain accurate results, we maintain the micro-blog text stream under the same period (one hour).

5 Evaluation

5.1 Experimental Setup

Dataset: A 3.5 million Sina micro-blogging dataset is adopted in this study to evaluate the proposed burst event detection system. Table 1 gives the date distribution of this dataset (from 1th September 2013 to 15th September 2013).

Table 1. Distribution of the dataset (by date)

<i>Date</i>	<i>01-02</i>	<i>03-04</i>	<i>05-06</i>	<i>07-08</i>	<i>09-10</i>	<i>11-12</i>	<i>13-14</i>	<i>15</i>
Num(10K)	712.2	706.6	768.3	752.2	749.4	759.1	698.5	366.7

Hardware: To evaluate our system, we construct a computer cluster. The configuration of each machine in this cluster is as follows: 64 bits CPU, 8 Kernels, 3.1GHz, 32GB RAM, and 1TB Hard disk. There are 10 machines in this cluster in which one is set as zookeeper which is also known as cluster control center while the others run Storm Topology.

5.2 Timeliness Evaluation

The first experiment evaluates the timeliness of RBEDS system. An example event detection result is shown in Figure 3.

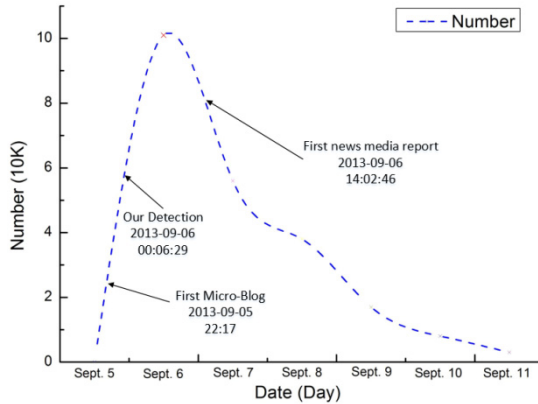


Fig. 3. Timing evaluation of the event “Lee Kaifu had the cancer”

As shown in figure 3, the event “李开复患癌 (Lee Kaifu had the cancer)” starts from Lee Kaifu Sina micro-blogging at 22:17 in 5th September, 2013. The original message is “世事无常，生命有限。原来，在癌症面前，人人平等。(The affairs of world are inconsistent, and life is finite. Originally, being face of cancer, everyone is equal)”. This event attracted many attentions. The communication peak occurred around 11 o'clock, 6th September. Then, it cooled down gradually. RDBEDMS system detected this burst event on 00:06:29, Sep.6, 2013, namely two hours later after its first occurrence. It is about 9hr ahead of the communication peak in Sina micro-blogging and 14hrs ahead of first media report of this event. This result shows the timeliness effectiveness of RBEDS.

Table 2. Timeliness evaluation of different categories

Category	Before peak	After peak
Social	26	7
Entertainment	47	20
Others	78	138

More experiments are performed on different types of events. The corresponding timeliness evaluation results are listed in Table 2. Table 2 gives the numbers of events detected by our system before they reached the peak or after the peak.

Where *Before peak* means the number of burst event detected before peak, *After peak* means the number of burst event detection after peak, respectively. It is shown that RBEDS are good at detecting social and entertainment event in short timeline while for other events, such as financial event, the detect timeliness performance is

lower. It attributes to the fact that social and entertainment events are always the hottest topic in people’s daily life. Serving as a self-media, the micro-blogging provides more obvious features of burst event detection in these topics. As for the other kind of events, the less people participation leads to the hysteresis of burst event detection.

5.3 Effectiveness Evaluation

The second experiment evaluates the effectiveness of RBEDS. The TDT evaluation metric [16] is employed here to evaluate the effectiveness. Detect error cost, C_{det} , measures the performance from two aspects: system omission ratio and false alarm ratio. It is defined as follows:

$$C_{Det} = C_{Miss} P_{Miss} P_{Target} + C_{FA} P_{FA} P_{NonTarget} \tag{2}$$

Where P_{Miss} is system omission ratio and P_{FA} is false alarm ratio. System omission ratio is the percentage of the new topics that the system failed to detect. The false alarm ratio is the percentage of wrongly detected event. C_{Miss} is cost coefficient of omission ratio, C_{FA} is false alarm ratio. P_{Target} is the prior probability that a story is a topic. $P_{Target} = 0.02$ is derived from training data, and $P_{NonTarget} = 1 - P_{Target}$.

The normalized detecting error cost C_{det} is defined as follows:

$$Norm(C_{Det}) = \frac{C_{Det}}{\min(C_{Miss} P_{Target}, C_{FA} P_{NonTarget})} \tag{3}$$

In this experiment, the training dataset is divided into three parts where each part consists of the micro-blogging for five days. Since Sina micro-blogging portal only provides daily hot topics, we manually annotate the events in each period and use the annotated data as validation set to compute the indices. In the stage of annotating data, we take $m=2$ and $m=3$. The obtained results are listed in Table 3 as follows:

It is observed that, the cost of detecting error cost $Norm(C_{Det})$ is obviously greater while the total false alarm ratio of while $m=3$ is smaller than $m=2$. When $m=3$, omission ratio increases obviously. Therefore, in real applications, $m=2$ is empirically selected.

Table 3. Evaluation on effectiveness

m	Group	$P_{Miss}(\%)$	$P_{FA}(\%)$	$Norm(C_{Det})$
1	1	22.34	3.91	0.41499
	2	23.57	3.72	0.41798
	3	21.04	3.83	0.39807
3	1	27.83	3.26	0.43804
	2	28.52	3.46	0.45474
	3	25.83	3.37	0.42343

5.4 Scalability Evaluation

The third experiment compares the efficiency of a single computer (pseudo-distributed) and computer cluster in order to evaluate the scalability of our system. The results of executing time of different nodes are shown in Table 4 as follows:

Table 4. The executing time of different nodes

<i>Dataset Size</i>	<i>Number of Nodes</i>	<i>Executing Time (Minutes)</i>
100MB	1	23
	4	7
	8	5
200MB	1	39
	4	13
	8	10

As shown in table 4, with the increase of the nodes number, the running time on the same dataset declines consequently. This result shows that our system can process different size data by changing the size of the cluster, namely, the system is scalable. In addition, due to the Storm has a strong fault-tolerant mechanism, when a cluster node failure, the system may recover without data loss.

6 Conclusion

This paper presents a real-time burst event detection system, named RBEDS, for handling large micro-blogging streams. The text-centric burst event detection approach and burst feature-centric approach are incorporated in this system. They are deployed under the Storm framework. The results of burst words detection and text clustering are incorporated to identify the burst event. Evaluations on 3.5-million Sina micro-blogging show that RBEDS detects burst event timely and effectively. Meanwhile, it is highly scalable. These results show that the RBEDS is able to process the full text stream of “real-world” Sina micro-blogging.

Acknowledgements. This study was supported by the National Natural Science Foundation of China (No. 61300112, 61370165), Natural Science Foundation of Guangdong Province (No. S2012040007390, S2013010014475), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, MOE Specialized Research Fund for the Doctoral Program of Higher Education 20122302120070, Open Projects Program of National Laboratory of Pattern Recognition, Shenzhen International Co-operation Research Funding GJHZ20120613110641217, Shenzhen Foundational Research Funding JCYJ20120613152557576 and JC201005260118A.

References

1. Allan, J.: Topic detection and tracking. Springer (2002)
2. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic Detection and Tracking Pilot Study: Final Report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218 (1998)
3. Brants, T., Chen, F., Farahat, A.: A System for New Event Detection. In: Proceedings of SIGIR (2003)
4. Diao, Q.M., Jiang, J., Zhu, F.D.: Finding Bursty Topics from Microblogs. In: Proceedings of ACL, pp. 536–544 (2012)
5. Wang, X.H., Zhai, X.X., Hu, X., Sproat, R.: Mining Correlated Bursty Topics Patterns from Coordinated Text Streams. In: Proceedings of ACM SIGKDD, pp. 784–793 (2007)
6. Du, Y.Y., He, Y.X., Tian, Y., Chen, Q., Lin, L.: Microblog Bursty Topic Detection Based on User Relationship. In: Proceedings of the 6th IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 260–263 (2011)
7. Du, Y.Y., Wu, W., He, Y.X., Liu, N.: Microblog Bursty Feature Detection Based on Dynamics Model. In: Proceedings of the International Conference on Systems and Informatics (ICSAI), pp. 2304–2308 (2012)
8. Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.J.: Parameter Free Burst Events Detection in Text Streams. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 181–192 (2005)
9. Wei, X., Zhu, F.D., Jing, J., Lim, E.P.: TopicSketch: Real-time Bursty Topic Detection from Twitter. In: Proceedings of IEEE International Conference on Data Mining (2013)
10. Bertino, E., Tan, K.L., Ooi, B.C., Sacks-Davis, R., Zobel, J., Shidlovsky, B., et al.: Indexing Techniques for Advanced Database System. Kluwer Academic Publishers (1997)
11. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: Proceedings of OSDI (2004)
12. Babcock, B., Bahu, S., Datar, M., Motwani, R., Widom, J.: Models and Issues in Data Stream System. In: Proceedings of SIGMOD/PODS (2002)
13. Brito, A., Martin, A., Knauth, T., Creutz, S., Becker, D., Weigert, S., Fetzer, C.: Scalable and Low-latency Data Processing with Stream MapReduce. In: Proceedings of CloudCom (2011)
14. Neumeyer, L., Robbins, B., Nair, A., Kesari, A.: S4: Distributed Stream Computing Platform. In: Proceedings of ICDMW (2010)
15. McCreadie, R., Macdonald, C., Ounis, I., Osborne, M., Petrovic, S.: Scalable Distributed Event Detection for Twitter. In: Proceedings of Big Data (2013)
16. Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., Amstutz, P.: Taking Topic Detection from Evaluation to Practice. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) – Track 4 (2005)

A Causative Attack Against Semi-supervised Learning

Yujiao Li^(✉) and Daniel S. Yeung

School of Computer Science and Engineering, South China University of Technology,
Guangzhou 510006, China
1160722938@qq.com, danyeung@ieee.org

Abstract. Semi-supervised learning plays an important role in pattern classification as it learns from not only the labeled sample but also the unlabeled samples. It saves the cost and time on sample labeling. Recently, semi-supervised learning has been applied in many security applications. An adversary may present in these applications to confuse the learning processes. In this paper, we investigate the influence of the adversarial attack on the semi-supervised learning. We propose a causative attack, which injects the attack samples in the training set, to mislead the training of the semi-supervised learning. The experimental results show the accuracy of the classifier trained by the semi-supervised learning drop significantly after attacking by our proposed model.

Keywords: Machine learning · Semi-supervised learning · Causative attack

1 Introduction

Machine learning can be divided into supervised learning, semi-supervised and un-supervised learning [1, 2]. A labeled set is given in a supervised learning. Mistakes of a learning process can be calculated. In un-supervised learning, there is no specific teacher [3]. The hidden structure between the samples are discovered. Differently, the semi-supervised learning is the combination of the supervised and the un-supervised learning [4, 5]. Only a partial set of samples are labeled because a label is costly and time-consuming in some applications, e.g. medical problems.

Nowadays, machine learning is widely used in security applications, e.g. email filtering [6, 7] and malware detection [8]. An adversary may present in these security application to mislead decision of the system by manipulate the samples. As the performance of traditional machine learning methods may downgrades significantly as they assume the training and testing set follow the same (but unknown) distribution.

Recently, many studies focus on the supervisor learning in an adversarial environment [9]. Many different methods, including robust learning [9] and data sanitization, with promising results have been proposed. However, there are only a few discussions on the security issue of unsupervised and semi-supervised learning [1, 2].

In this paper, we investigate the robustness of the semi-supervised learning under an adversarial attack. As semi-supervised learning contains less information than supervised learning (i.e. only some samples are labeled), semi-supervised learning may

be more easily to be attacked. We propose a causative attack aiming to increase the error rate on the malicious samples to self-training method in semi-supervised learning. We assume the attacker obtains the information of the classifier, and labeled and unlabeled samples. The feature values of unlabeled samples located near to the final decision plane are moved closer to the legitimate class. Experiment is carried out to evaluate the performance of the proposed attack algorithm.

The rest of this paper is organized as follows. The related work of adversarial learning and semi-supervised learning is described in section 2. Section 3 mentions the proposed attack model to the self-training method. The experimental results are reported in Section 4 and finally, the conclusion is given in section 5.

2 Background

In this section, the related work of this paper is introduced briefly. The background of adversarial learning and the semi-supervised learning are described in section 2.1 and 2.2 respectively.

2.1 Adversarial Learning

An adversary who intentionally manipulates the samples in order to mislead the decisions of a classifier presents in an adversarial learning problem [9, 10, 11]. For example, the bad word in spam, which appear frequently in spam but not in legitimate message, is camouflaged to avoid the detection of spam filtering (e.g. Viagra is changed to Viagr@). The performance of traditional learning algorithms may drop significantly under the adversarial attack as they do not consider the change of the data distribution.

Adversarial attacks can be separated into two types, such as the causative and exploratory attacks [11]. The testing samples are manipulated in order to evade the detection of a classifier in exploratory attack. For instance, the good (bad) words are inserted to (removed from) a spam message in evasion attack [13]. As a result, the camouflaged spam message can pass through the spam filter data and be sent to users. On the other hand, causative attack poisons the training set by changing the feature values [9] or flipping the labels [9] to mislead the learning of a classifier. The generalization ability of the classifier is low as the testing samples are different from the attacked training samples. For example, fake attack packets are generated by red hering attack [12].

Causative attack against classifier is more difficult than exploratory attack, because it needs more strong assumptions. In addition, in supervised learning, the label of training samples is determined at the beginning, it is difficult to be changed. But in semi-supervised learning, there are a lot of unlabeled samples in the training set, so the attacker can attack it easier than supervised learning. We will introduce our causative attack against semi-supervised learning in section 3.

2.2 Semi-supervised Learning

Semi-supervised learning is a hot spot at present. It mainly combines several of labeled instances and a lot of unlabeled instances to train a classifier.

Because semi-supervised learning needs to use unlabeled samples, it needs some assumption to determine the distribution of the dataset. There are three main assumptions, smoothness assumption, cluster assumption and manifold assumption [5]. Semi-supervised learning algorithms make use of at least one of these assumptions. Smoothness assumption mentions if two samples are close together, they may have the same label. It mainly fits for simple decision boundaries. However, this assumption may not be held when the samples are in low-density space. Cluster assumption describes if a dataset can be separated into some discrete clusters, the samples in the same cluster may be in the same category. In this case, a sample may be located in more than one cluster. Cluster assumption is a special case of the smoothness assumption. Based on manifold assumption, the dimension can be reduced if the input data is in high dimension. The manifold assumption is effective when data is high-dimensional which is difficult to model directly, but which only has a few degrees of freedom [5].

Generative models [4] which calculate a probability density function [17] or a conditional probability density function [18] to determine the class of a sample, is one of semi-supervised learning methods. Another example is S3VMs [4] (also named as TSVM). This method maximizes the margin of the hyperplane by assigning different labels on unlabeled samples. Graph-based algorithms use a graph to represent the structure of a dataset. A node in the graph denotes the labeled or unlabeled sample. A weight on an edge represents the similarity between the nodes. The label is determined according to the similarity. Multi-view algorithms, e.g. co-training [20, 21], use two or more classifiers to decide the label of the unseen samples by majority voting [22]. The algorithms assume that the different views are sufficient and redundant. A classifier is trained iteratively in self-training algorithm method. The most confident unlabeled sample(s) based on the current classifier is chosen and assigned the label. The algorithm stops until the unlabeled set is null.

An example of the self-training methods is Naïve Bayes algorithms [23], which is focused in this paper. Let LS_i and ULS_i be the labeled and unlabeled set in the i^{th} iteration, and C be the set of classes containing + and -. $i=0$ represents the initial state. A feature vector $x = [x_1, x_2, \dots, x_m]$ where m is the number of features and $x \in X$. $\langle x, c \rangle$ denotes the sample and c is either the true label for the labeled sample or predicted label for unlabeled sample. The function f defines as $X \rightarrow C$. By assuming the features of a sample x is conditionally independent, the Naïve Bayes classifier f can be define as:

$$\begin{aligned}
 f(x) &= \arg \max_{c_j \in C} P(c_j | x) \\
 &= \arg \max_{c_j \in C} \frac{P(c_j) * P(x | c_j)}{P(x)} \\
 &= \arg \max_{c_j \in C} \frac{P(c_j) * \prod_i^n P(f_i | c_j)}{P(x)}
 \end{aligned} \tag{1}$$

The class distribution entropy [8], defined as (2), is applied to measure the uncertainty of x . If the entropy is close to 0, there is more confident that the decision of a classifier is correct. As a result, the sample is located far away from the decision boundary. Otherwise, the sample is located near to the boundary.

$$E(x) = -\sum_{c_j \in C} (P(c_j|x) * \log(P(c_j|x))) \quad (2)$$

The self-training algorithm shows as follow [18]. An output of a classifier f which is trained by using LS and each sample in ULS . p number of the samples which have the lowest entropy is moved from ULS to LS with the predicted label. The process repeats until ULS is null.

Algorithm 1: Self-Training Algorithm

Input: LS , the labeled set; ULS , the unlabeled set; i , the iteration.

Output: the predicted label of each sample in ULS

1: $i \leftarrow 0$;

2: Train a classifier f by LS_i ;

3: *Repeat*

4: Predict $f(x)$ for each x in ULS_i ;

5: Select p samples with the lowest entropy from ULS_i as A ;

6: Add the samples in A with their predicted label into set LA ;

7: Add LA and LS_i into LS_{i+1} and delete A from ULS_i as ULS_{i+1} ;

8: Set $A = \emptyset$ and $LA = \emptyset$;

9: $i \leftarrow i + 1$;

10: *Until* $ULS_i = \emptyset$;

3 Proposed Attack Model

The proposed attack model aims to reduce the accuracy of the self-training algorithm of the semi-supervised learning on the malicious samples. We assume the adversary has the knowledge on the classifier, the labeled and the unlabeled set of the user. The attack ability is limited to manipulating the features of the samples in unlabeled set. Next section introduces a simple sample to illustrate the motivation of the model and then the detail procedure will be devised.

3.1 Motivation

5000 and 1000 samples from the dataset of 1999 KDD intrusion detection contests [24] are chosen as the training and testing set. We choose 50 samples randomly in the training set as the initial labeled set LS and the rest are the unlabeled set ULS . Only malicious samples are selected for the testing set as the attack model focus on the accuracy on the malicious samples. Figure 1 illustrates the accuracy values of the classifier are stably around 80.5% before the 4200th iteration. Then, the accuracy increases rapidly to 92.5% from 4200th to 5000th iteration. It shows the samples selected from ULS are more informative for training from these iterations, i.e. the decision boundary is determined significantly by those samples. As a result, we propose an

attack method which modifies the samples affecting the decision boundary in order to increase the efficiency of the attack. The detail is given in next section.

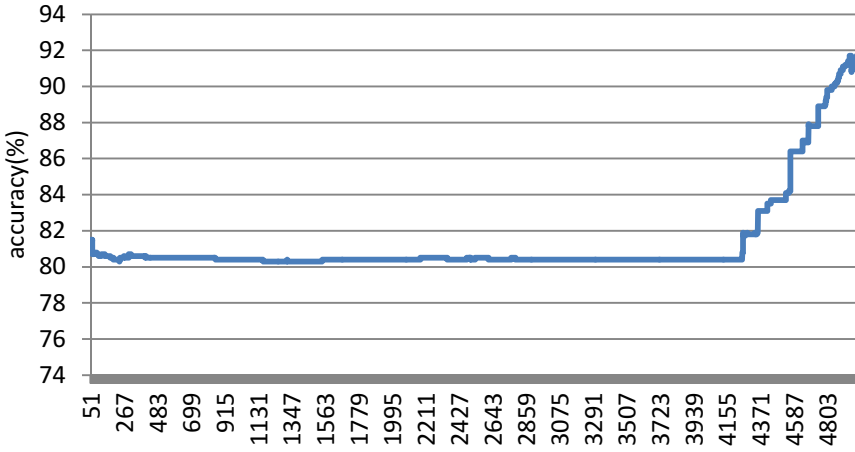


Fig. 1. Accuracy of each iteration of self-training method

3.2 Attack Algorithm

The proposed attack model manipulates the samples which significantly affect the decision plane, as mentioned in previous section. These samples are identified according to the accuracy of the surrogate function trained by using the given training set. $iter^{th}$ iteration is determined when the accuracy of the classifier increases sharply. All samples which are left in the unlabeled set after $iter^{th}$ iteration are attacked. Then, each attack sample is moved toward to the legitimate samples with the ratio $lamda$. The range of $lamda$ is from 0 to 1. After that, the unlabeled set is attacked.

Algorithm 2:Generate Fake Samples Algorithm

Input: $iter$, change the unlabeled samples added into LS after $iter^{th}$ iteration; LS_{iter} , the $iter^{th}$ labeled set; ULS_{iter} , the $iter^{th}$ unlabeled set; ulx_{iter} , each sample in ULS_{iter} ; lx_{iter} , each sample in L_{iter} .

Output: the malicious samples MS

1: Find the centralpoint of the samples with the label $c = -1$ in L_{iter} as a vector \overrightarrow{midN} , label $c=+1$ in L_{iter} as \overrightarrow{midP} , and in ULS_{iter} as $\overrightarrow{midULS_{iter}}$;

2: $\overrightarrow{\delta} \leftarrow \overrightarrow{midULS_{iter}} - \overrightarrow{midN}$;

3: Repeat

4: $ulx'_{iter} \leftarrow ulx_{iter} + lamda * \overrightarrow{\delta}$;

5: Add ulx'_{iter} into MS and delete ulx_{iter} from ULS_{iter} ;

6: Until $ULS_{iter} = \emptyset$;

4 Experiment

The dataset of 1999 KDD intrusion detection contests [24] is used in the experiment. Each sample contains 9 discrete features and 32 continuous features. All the attacked types are grouped as a positive class ($c=+1$) and the normal type is the negative class ($c=-1$). 6000 samples are randomly selected as our training set including 1000 samples as testing set (TS) and the rest form a training set. The features are removed if they contain the same values for all samples. As a result, 35 features are used. 50 samples from training set are randomly selected as LS initially and the rest are ULS . Each time only one sample is added to LS from ULS . As the objective of the attack is to evade the detection on the malicious samples, the false negative (FN) is considered. The proposed attack model focuses on Naïve Bayes algorithm. Each experiment has been repeated 3 times independently.

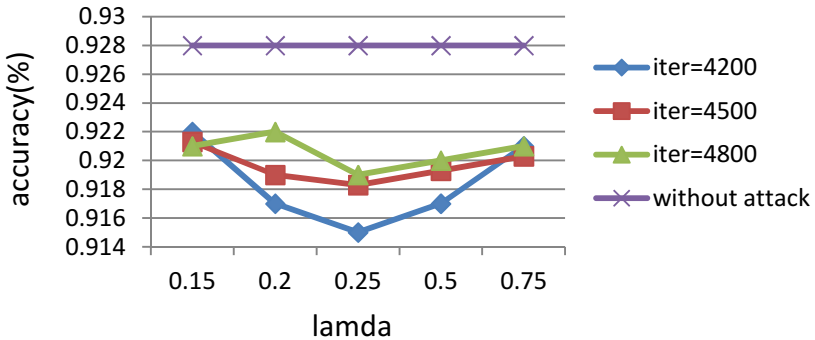


Fig. 2. Accuracies of a classifier using Naïve Bayes algorithm under the proposed attack strategy with different attack strengths

Figure 2 shows the average accuracies of classifiers using Naïve Bayes algorithm without and with attack by our proposed model with different $lamda$ values and starting iterations ($iter$). The accuracy of semi-supervised classifier without attack is about 92.8%, which is higher than all cases under an attack. It indicates the proposed attack model downgrades the performance of the classifier. It is worth mentioning the feature noise attack without considering the label of a sample is not efficient, i.e., the accuracy of a classifier decreases by manipulating many samples. The accuracy drops about 1.5% in [8] by modifying 100000 samples.

The result also illustrates that a smaller starting iteration number reduces the accuracy. When $iter = 4200$, the attack model reduces the accuracy values the most. It may be because that more samples are modified when $iter = 4200$ than the ones in other cases. Moreover, the classifier has the worst performance when $lamda$ is 0.25. If $lamda$ is large, the samples in ULS will be pushed into class -, in which the boundary will not be changed. On the other hand, the locations of ULS do not moved significantly to affect the boundary when $lamda$ is small. It explains why $lamda$ value should not be too small or too large.

5 Conclusion

This paper purposes a causative attack strategy to the self-training algorithm with Naïve Bayes algorithm in semi-supervised learning problem. Only the samples which significantly affect the learning in the unlabeled set are moved closer to the legitimate class by modifying their feature values. Experimental results illustrate the method reduces the accuracy of the classifier. Moreover, it also suggests that less samples are modified according to the proposed algorithm to achieve the same performance in comparison with the existing method.

References

1. Mitchell, T.M., Zeng, J.-H. (trns.): *Machine Learning*, pp. 2–5. China Machine Press (2003)
2. Mohri, M., Rostamizadeh, A., Talwalkar, A.: *Foundations of Machine Learning*, pp. 1–14. MIT Press (2012)
3. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, pp. 517–518. Wiley-InterScience (2000)
4. Zhu, X.-J., Goldberg, A.B.: *Introduction to Semi-Supervised Learning*, pp. 1–130. Morgan & Claypool Publishers (2009)
5. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*, pp. 119–134. MIT Press (2006)
6. Zhou, Y., Jorgensen, Z., Inge, M.: Combating Good Word Attacks on Statistical Spam Filters with Multiple Instance Learning. In: *ICTAI*, pp. 298–305 (2007)
7. Wang, X.-W., Wang, Z.-F.: Good Word Attack Spam Filtering Model Based On Artificial Immune System. In: *ACAI*, pp. 1106–1109 (2012)
8. Zhu, F., Long, J., Zhao, W., Cai, Z.: A Misleading Attack against Semi-supervised Learning for Intrusion Detection. In: Torra, V., Narukawa, Y., Daumas, M. (eds.) *MDAI 2010*. LNCS, vol. 6408, pp. 287–298. Springer, Heidelberg (2010)
9. Nelson, B.: *Behavior of Machine Learning Algorithms in Adversarial Environments*, Phd Dissertation, pp. 37–55 (2010)
10. Lowd, D., Meek, C.: Adversarial Learning. In: *ACM SIGKDD*, pp. 641–647 (2005)
11. Nelson, B., Biggio, B., Laskov, P.: Understanding the Risk Factors of Learning in Adversarial Environments. In: *ACM Workshop on Artificial Intelligence and Security*, pp. 87–92 (2011)
12. Newsome, J., Karp, B., Song, D.: Poly graph: Automatically generating signatures for polymorphic worms. In: *S&P IEEE Symposium*, pp. 226–241 (2005)
13. Zhou, X.-C., Shen, H.-B., Huang, Z.-Y., Li, G.-J.: Large margin classification for combating disguise attacks on spam filters. *Journal of Zhejiang University SCIENCE C*, 155–238 (2012)
14. Luo, R.C., Lin, P.-H., Wu, Y.-C., Huang, C.-Y.: Dynamic face recognition system in recognizing facial expressions for service robotics. In: *AIM*, pp. 879–884 (2012)
15. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: *ICML*, pp. 200–209 (1999)
16. Zhou, Z.-H.: Co-Training with Insufficient Views. In: *ACML*, pp. 467–482 (2013)
17. Ushakov, N.G.: Density of a probability distribution, pp. 313–333. *Encyclopedia of Mathematics*, Springer (2001)
18. Gut, A.: *Probability: A Graduate Course*, pp. 113–122. Springer (2005)

19. Bishop, C.M., Lasserre, J.: Generative or Discriminative? getting the best of both worlds. *Bayesian Statistics* **8**, 3–23 (2007)
20. Mao, C.H., Lee, H.M., Parikh, D., Chen, T., Huang, S.Y.: Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In: *ACM SAC*, pp. 2042–2048 (2009)
21. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: *COLT*, pp. 92–100 (1998)
22. Zhou, Z.-H., Li, M.: Semi-Supervised Regression with Co-Training. In: *IJCAI*, pp. 1479–1493 (2007)
23. Qin, Z.-C.: Naive Bayes Classification Given Probability Estimation Trees. In: *ICMLA*, pp. 34–42 (2006)
24. Archive, T.U.K.: Kdd cup 1999 data (1999)

Decision Tree Learning

Study and Improvement of Ordinal Decision Trees Based on Rank Entropy

Jiankai Chen, Junhai Zhai^(✉), and Xizhao Wang

Machine Learning Center, Faculty of Mathematics and Computer Science,
Hebei University, Baoding 071002, China
mczjh@hbu.cn

Abstract. Decision tree is one of the most commonly used methods of machine learning, and ordinal decision tree is an important way to deal with ordinal classification problems. Through researches and analyses on ordinal decision trees based on rank entropy, the rank mutual information for every cut of each continuous-valued attribute is necessary to determine during the selection of expanded attributes for constructing decision trees based on rank entropy in ordinal classification. Then we need to compare these values of rank mutual information to get the maximum which corresponds to the expanded attribute. As the computational complexity is high, an improved algorithm which establishes a mathematical model is proposed. The improved algorithm is theoretically proved that it only traverses the unstable cut-points without computing the values of stable cut-points. Therefore, the computational efficiency of constructing decision trees is greatly improved. Experiments also confirm that the computational time of the improved algorithm can be reduced greatly.

Keywords: Ordinal decision tree · Rank entropy · Unstable cut-point

1 Introduction

Decision tree [1] is a typical induction algorithm. The attributes of the training samples are divided into two kinds, symbols attributes and continuous attributes. The value domains of the symbols attributes are finite sets without ordering relation while the value domains of the continuous attributes are sets with ordering relation, such as the subsets of real or integer [2]. Quinlan presented the ID3 algorithm [3] in 1986, which is a famous decision tree algorithm which dealing with symbol attribute problems. Then Quinlan and Breima presented the C4.5 algorithm [4] and CART (Classification and Regression Tree) [5], which can handle symbolic value properties, and also continuous valued attributes. These algorithms basically compute the predictive ability of decision by using the information entropy.

Ordinal classification tasks widely exist in real world life and work. We select commodities in a market according to the price and quality; employers select their employees based on their education and experience; investors select stocks or bonds in terms of their probability of appreciation or risk; Universities select scholarship offers according to students' performances. Editors make a decision on a manuscript

according to its quality. These involve ordering relation, but information entropy cannot reflect the ordering relation. In order to deal with many decision making tasks [6, 7, 8, 9], Ben-David et al have developed ordinal decision trees, monotonic decision trees and rank decision trees respectively [10, 11, 12, 13, 14], but the generalization is not well enough. Hu presented the conceptions of rank entropy and rank mutual information [15] and building ordinal decision trees based on rank mutual information [16]. The algorithm can reflect the order structure information very well, but it needs to compute the rank mutual information of each cut for each of the continuous-valued attributes during the selection of expanded attributes for learning of decision trees in ordinal classification, the computational complexity is very high.

In this paper, we introduce the concepts of unstable cut-points [2] and improve the algorithm of ordinal decision trees based on rank entropy. The experiments and theory prove that the improved algorithm can reduce computational complexity and improve the computational efficiency greatly.

2 The Relevant Knowledge of Generating Ordinal Decision Tree

Let $U = \{x_1, x_2, \dots, x_n\}$ be a set of samples, A be a set of attributes to describe the samples and D is a finite ordinal set of decisions. $v(x_i, a)$ and $v(x_i, D)$ denotes the value of x_i in attributes $a \in A$ and D , respectively. The ordinal relations between samples in terms of attribute a or D are denoted by \leq . We say x_j is no worse than x_i in terms of a or D if $v(x_i, a) \leq v(x_j, a)$ or $v(x_i, D) \leq v(x_j, D)$, denoted by $x_i \leq_a x_j$ and $x_i \leq_D x_j$, respectively. Correspondingly, we can also define $x_i \geq_a x_j$ and $x_i \geq_D x_j$. Given $B \subseteq A$, we say $x_i \leq_B x_j$ if for $\forall a \in B$, we have $v(x_i, a) \leq v(x_j, a)$. A predicting rule is a function [16].

$$f : U \rightarrow D ,$$

which assigns a decision in D to each sample in U . A monotonically ordinal classification function should satisfy the following constraint

$$x_i \leq x_j \Rightarrow f(x_i) \leq f(x_j), \forall x_i, x_j \in U .$$

Definition 1 rank entropy [15]

Given $DT = \langle U, A, D \rangle$, $B \subseteq A$. The ascending and descending rank entropies of the system with respect to B are defined as

$$RH_B^{\leq}(U) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\leq}|}{n} \tag{1}$$

$$RH_B^{\geq}(U) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\geq}|}{n} \tag{2}$$

Definition 2 rank mutual information [15]

Given $DT = \langle U, A, D \rangle$, $B \subseteq A, C \subseteq A$. The ascending rank mutual information (ARMI) and descending rank mutual information (DRMI) of the set U between B and C are defined as

$$RMI^{\leq}(B, C) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\leq} \times [x_i]_C^{\leq}|}{n \times |[x_i]_B^{\leq} \cap [x_i]_C^{\leq}|} \tag{3}$$

$$RMI^{\geq}(B, C) = -\frac{1}{n} \sum_{i=1}^n \log \frac{|[x_i]_B^{\geq} \times [x_i]_C^{\geq}|}{n \times |[x_i]_B^{\geq} \cap [x_i]_C^{\geq}|} \tag{4}$$

The ordinal decision tree algorithm based on rank mutual information is formulated as follows:

Input: criteria: attributes of samples.

Decision: decision of samples.

Stopping criterion: If the maximal rank mutual information is less than ϵ , the branch stops growing, or if the number of sample is 1 or all the samples come from the same class, the branch stops growing

Output: ordinal decision tree T.

Begin: generate the root node.

Step1 for (each attribute $A_i \in Criteria$)

Step2 for (each $c_j \in A_i$)

Step3 divide samples into two subsets according to c_j .

If $A_i(x) \leq c_j$, then $A_i(x) = 1$, else $A_i(x) = 2$. Compute

$$RMI(A_i, c_j) = RMI(Criteria, Decision)$$

Step4 select $c_j^*, c_j^* = \arg \max_j RMI(A_i, c_j)$

Step5 select the best feature and the corresponding

$$\text{splitting point: } (A, c^*) = \max_i RMI(A_i, c_j^*)$$

Step6 build a new node and split samples with A, c^* . recursively produce new splits according to step1-step5 until stopping criterion is satisfied.

End

3 Improvement of Ordinal Decision Tree

The computational complexity of the above algorithm is very high when each attribute of the training set has distinct values. To solve this problem, an improved algorithm is given in this section.

3.1 Application of Stable and Unstable Cut-Points

Definition 3 stable and unstable cut-points

Given $DT = \langle U, A, D \rangle$, The samples are first sorted by increasing value of the attribute A_i . Supposing that it is indicated with $x_j (j = 1, 2, 3, \dots, N)$ after the sorting, the mid-value of x_j and x_{j+1} on the attribute A_i is called cut-points of the attribute A_i , denoted by $c_{ij} (1 \leq j \leq N)$. If the classes of the sample x_j and x_{j+1} are different, then c_{ij} is called unstable-points, otherwise it is called stable-points.

This paper still selects the rank mutual information as the index in constructing ordinal decision trees, then divide cut-points to the cut-points stable-points and unstable-points. The below proves that the rank mutual information function must achieve its maximum in unstable cut-points.

3.2 Proposition Proof

Proposition. If the cut-point T maximizes the rank mutual information $RMI^{\leq}(A, D)$, then T is an unstable-point.

Proof: In the training set U , assuming that the number of classes set is two (that is 1 and 2) and the number of the samples is N . The samples $x_j (1 \leq j \leq N)$ are first sorted by increasing value of the attribute A_i , and the midpoint between each successive pair of examples in the sorted sequence is evaluated as a potential cut point. Then taking two points of which the class is same and the distance is closest, denoted by C_1 and C_2 . The midpoint between the point C_1 and the point of the right adjacent with C_1 is denoted by unstable cut-point B_1 . In the same way, the midpoint between the point C_2 and the point of the right adjacent with C_2 is denoted by unstable cut-point B_2 . T is a cut-point in (B_1, B_2) . Figure 1 illustrates this situation.

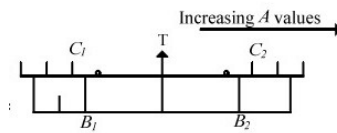


Fig. 1. Cut point instructions

Assuming that there are $m-1$ stable-points between B_1 and B_2 , the m samples of the interval (B_1, B_2) are the same classes. Let the class of the m samples is 1, then the class of C_1 and C_2 is 2. A variable x is introduced, which indicates the x stable-points ($1 \leq x \leq m$). In figure 2, let the set of the samples with $A \leq C_1$ is denoted by U_L , the set of the samples with $A \geq C_2$ is denoted by U_R . In the set U_L , the number of the samples

with 1, 2 classes is L_1 and L_2 respectively. In the set U_R , the number of the samples with 1,2classes is R_1 and R_2 respectively. They satisfy the conditions :

$$0 \leq L_1 + L_2, R_1 + R_2 \leq N - m, L_1 + L_2 + R_1 + R_2 + m = N$$

The cut-point T divides the range of the attribute A to two sets: $U_1 (A \leq T)$, $U_2 (A > T)$. According to the algorithm step3 of the ordinal decision tree based on rank entropy, it knows that the value of attribute A of the set U_1 is 1 and the value of attribute A of the set U_2 is 2. The rank mutual information $RMI^{\leq}(A, D)$ is selected to measure the uncertainty when T ranges of the interval (B_1, B_2)

$$RMI^{\leq}(A, D) = -\frac{1}{|U|} \sum_{i=1}^N \log \frac{|[x_i]_A^{\leq}| \times |[x_i]_D^{\leq}|}{|U| \times |[x_i]_A^{\leq} \cap [x_i]_D^{\leq}|}, (x_i \in U) \tag{5}$$

When $i = 1 : (L_1 + L_2 + x - 1)$, we get

$$RMI_1^{\leq} = -\frac{1}{|U|} \sum_i^{L_1+L_2+x-1} \log \frac{|[x_i]_A^{\leq}| \times |[x_i]_D^{\leq}|}{N \times |[x_i]_A^{\leq} \cap [x_i]_D^{\leq}|} = -\frac{1}{|U|} \sum_i^{L_1+L_2+x-1} \log \frac{N \times |[x_i]_D^{\leq}|}{N \times |[x_i]_D^{\leq}|} = 0$$

When $i = (L_1 + L_2 + x) : (L_1 + L_2 + m)$,

$$RMI_2^{\leq} = \frac{1}{|U|} \sum_{i=L_1+L_2+x}^{L_1+L_2+m} \log \frac{|[x_i]_A^{\leq}| \times |[x_i]_D^{\leq}|}{N \times |[x_i]_A^{\leq} \cap [x_i]_D^{\leq}|} = \frac{1}{N} \left[(m+1-x) \log \frac{(m+R_1+R_2+1-x) \times (m+L_2+R_2)}{N \times (m+R_2+1-x)} \right]$$

When $i = (L_1 + L_2 + m + 1) : N$,

$$RMI_3^{\leq} = \frac{1}{|U|} \sum_{i=L_1+L_2+m+1}^N \log \frac{|[x_i]_A^{\leq}| \times |[x_i]_D^{\leq}|}{N \times |[x_i]_A^{\leq} \cap [x_i]_D^{\leq}|} = \frac{1}{N} \left[R_1 \log \frac{(m+R_1+R_2+1-x) \times (m+L_2+R_2)}{N \times (m+R_2+1-x)} \right]$$

To sum up:

$$RMI^{\leq}(A, D) = RMI_1^{\leq} + RMI_2^{\leq} + RMI_3^{\leq}$$

$$= -\frac{1}{N} \left[(m+R_2+1-x) \log \frac{(m+R_1+R_2+1-x) \times (m+L_2+R_2)}{N \times (m+R_2+1-x)} \right], (1 \leq x \leq m)$$

Let

$$F(x) = -\frac{1}{N} \left[(m+R_2+1-x) \log \frac{(m+R_1+R_2+1-x) \times (m+L_2+R_2)}{N \times (m+R_2+1-x)} \right], (1 \leq x \leq m)$$

The first order partial derivative:

$$\frac{dF(x)}{dx} = \frac{1}{N} \left[\frac{R_1}{(m+R_1+R_2+1-x)} - \log \frac{(m+R_1+R_2+1-x) \times (m+L_2+R_2)}{N \times (m+R_2+1-x)} \right], (1 \leq x \leq m)$$

The second order partial derivative:

$$\frac{d^2F(x)}{dx^2} = \frac{R_1^2}{N \times (m + R_1 + R_2 + 1 - x)^2 (m + R_2 + 1 - x)}, \quad (1 \leq x \leq m)$$

m, R_1, R_2 are greater than zero, since $\frac{d^2F(x)}{dx^2} > 0$, $F(x)$ is a concave function which can reach maximum on the boundary points. That is, $RMI^{\leq}(A, D)$ is maximize at $x = 1$ or at $x = m$ thus forcing T to coincide with one of the boundary points B_1 or B_2 .

This proves that the maximum value of $RMI^{\leq}(A, D)$ must therefore occur at one of the two unstable cut-points B_1 or B_2 .

3.3 Induction of Improved Algorithm

This paper theoretically proves that the rank mutual information function will not achieve its maximum in stable cut-points, but only in unstable cut-points. This result means that the improved algorithm only traverses its unstable cut-points without computing the value of stable cut-points. The induction of improved algorithm is as follows:

Input: criteria: attributes of samples.

Decision: decision of samples.

Stopping criterion: If the maximal rank mutual information is less than ϵ , the branch stops growing, or if the number of sample is 1 or all the samples come from the same class, the branch stops growing

Output: ordinal decision tree T.

Begin: generate the root node.

Step1 for (each attribute $A_i \in Criteria$)

Step2 we do a sort for these samples $x_j (1 \leq j \leq N)$ from small to large based on attributes A_i , then

search for all unstable-points $t_j (j=1,2,3...k)$ after the sorting

Step3 for (each $t_j \in A_i$)

Step4 divide samples into two subsets according to t_j

If $A_i(x) \leq t_j$, then $A_i(x) = 1$, else $A_i(x) = 2$. Compute $RMI(A_i, c_j) = RMI(Criteria, Decision)$

Step5 select $t_j^*, t_j^* = \arg \max_j RMI(A_i, t_j)$

Step6 select the best feature and the corresponding splitting point: $(A, t^*) = \max_i RMI(A_i, t_j^*)$

Step7 build a new node and split samples with A, t^* . recursively produce new splits according to step1-step6 until stopping criterion is satisfied.

End

4 Experimental Results and Analysis

4.1 The Experimental Results Before and After Introducing Unstable Cut-Points

In order to show the effectiveness of the improved algorithm, we conduct some numerical experiments with artificial and real-world datasets

First the following function is introduced to generate monotone datasets:

$$f(x_1, x_2) = 1 + x_1 + \frac{1}{2}(x_2^2 - x_1^2) \tag{6}$$

where x_1, x_2 are two random variables independently drawn from the uniform distribution over the unit interval. In order to generate ordered class labels, the resulting numeric values are discretized into 2 intervals $[0, 1/2], [1/2, 1]$. Then it forms a 2-class monotonic classification task. The datasets are given table1.

In the experiments process, we recorded the number of the original candidate cut-points and unstable cut-points and the time of building ordinal tree respectively, then make a detailed comparison, which is given table2, table3.

Table 1. Artificial data

Datasets	Number of instances	Attributes	Class
Data 1	50	2	2
Data 2	100	2	2
Data 3	300	2	2
Data 4	500	2	2
Data 5	600	2	2
Data 6	800	2	2
Data 7	1000	2	2

Table 2. Compare the number of cut-points

Datasets	Number of original candidate cut-points	Unstable cut-points	Cut-points loss
Data 1	100	47	53.00%
Data 2	200	68	66.00%
Data 3	600	204	66.00%
Data 4	1000	348	65.20%
Data 5	1200	410	65.83%
Data 6	1600	504	68.50%
Data 7	2000	638	68.10%

Table 3. Compare the time of building trees

Datasets	Time of the original algorithm(s)	Time of the improved algorithm(s)	Time loss
Data 1	0.16900	0.13700	18.60%
Data 2	0.20500	0.15300	25.24%
Data 3	0.90600	0.28400	68.84%
Data 4	0.645000	0.42300	34.78%
Data 5	0.71300	0.42500	40.70%
Data 6	1.050900	0.616100	41.39%
Data 7	1.192000	0.473000	60.37%

In order to test how the improved algorithm behaves in real-world applications, we collected 7 datasets. Four datasets come from UCI repository: Glass, House, CPU, Bodyfat. Before training decision trees, we have to preprocess the datasets. Because it uses ascending rank mutual information as the splitting rules. The datasets is given table4.

Table 4. Real-world datasets

Datasets	Number of instances	Attributes	Class
CPU	204	6	6
Bodyfat	252	14	2
House	506	11	2
Glass	214	9	2
Workers	63	25	2
Squash	50	24	3
Segment	497	14	2

We also recorded the number of the original candidate cut-points and unstable cut-points, and the time of building ordinal tree respectively, which are given table 5, table 6.

Table 5. Compare the number of cut-points

Datasets	Number of original candidate cut-points	Unstable cut-points	Cut-points loss
CPU	1218	219	82.02%
Bodyfat	3514	914	73.99%
House	5555	1202	78.36%
Glass	1917	616	67.87%
Workers	1550	610	60.65%
Squash	1176	648	44.90%
Segment	6944	1873	73.03%

Table 6. Compare the time of building trees

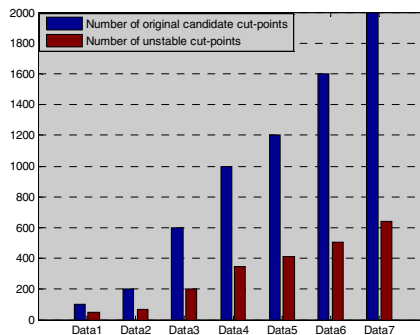
Datasets	Time of the original algorithm(s)	Time of the improved algorithm(s)	Time loss
CPU	2.67500	1.37900	48.33%
Bodyfat	4.61300	2.69000	41.73%
House	4.53000	2.37100	47.66%
Glass	1.63000	0.84400	47.64%
Workers	3.60600	1.09700	69.32%
Squash	0.73500	0.35500	51.91%
Segment	8.26000	4.01200	51.96%

4.2 Effective Analysis of the Improved Algorithm

Assuming that the number of the training set U with k classes is N , and the number of the attributes ($A_i (i = 1, 2, 3, \dots, j)$) is j . In the original algorithm, it must be evaluated N times for each attribute (assuming that the N samples have distinct values). In the case, the rank mutual information need to be computed $j \times N$ times for all attributes.

After introducing the unstable-points, these samples are first sorted by increasing value of the attribute A_i . Since the improved algorithm only traverses its unstable cut-points without computing the value of stable cut-points, the rank mutual information only need to be computed $j \times t$ times for all attribute, t satisfies such requirement: $k - 1 \leq t \leq N - 1$ and $k \ll N$. For large scale training samples of which unstable cut-points is relatively concentrated, the improved algorithm reduces the times of computing rank mutual information greatly and improves the learning effectiveness.

In order to illustrate the effectiveness of the improved algorithm, an analysis of the above experimental results is given. As shown in figure 2,3,4,5.

**Fig. 2.** The number of cut-point (artificial)

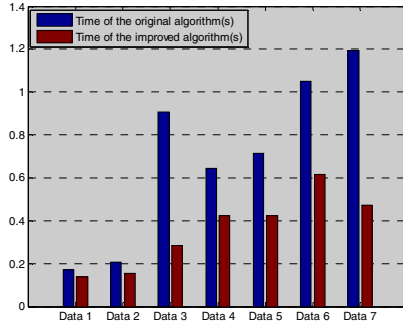


Fig. 3. The time of building tree (artificial)

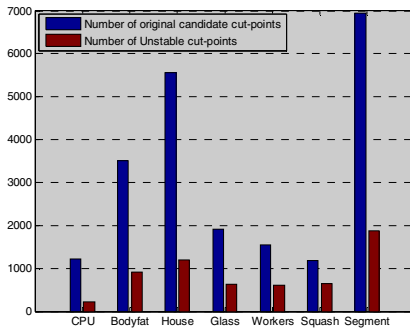


Fig. 4. The number of cut-points (real)

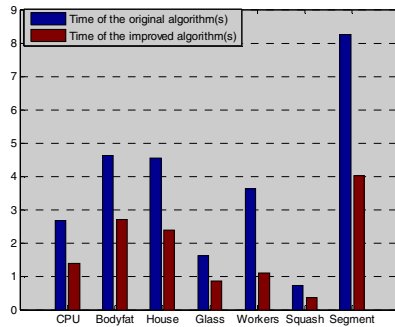


Fig. 5. The time of building tree (real)

Figure 2, 4 show the comparison of the number of the candidate cut-points in building trees with artificial data and real-world data, respectively. It indicates that the improved algorithm reduces computational complexity significantly. Figure 3, 5 show the comparison of the running time in building trees with artificial datasets and real-world datasets, respectively. It indicates that the improved algorithm reduces the running time significantly. The above two aspects verify the effectiveness of the improved algorithm.

5 Conclusions

It is a crucial problem to select expanded attributes for building ordinal decision trees. Since we need to compute the rank mutual information of each cut for each of the continuous-valued attributes during the selection of expanded attributes for learning of decision trees based on rank entropy in ordinal classification. When machine learning programs are designed to work on large sets of training data (especially, the samples have distinct values for each attribute), the computational complexity is high. In order to deal with this problem, this paper proposed and improved algorithm and a series of experiments with artificial datasets and real-world datasets are performed. The results showed that the improved algorithm reduced the computational complexity and improved the effectiveness significantly.

Acknowledgements. This research is supported by the National Natural Science Foundation of China (61170040), by the Natural Science Foundation of Hebei Province (F2012201023, F2013201110 and F2013201220), by the Key Scientific Research Foundation of Education Department of Hebei Province (ZD2010139), by the natural science foundation of Hebei University (2011-228) , by the research projects on reform of education and teaching of Hebei University (JX07-Y-27), and by Soft science research project of Hebei Province (12457662).

References

1. Mitchell, T.M.: Machine Learning, 1st edn. McGraw-Hill Science/Engineering/Math (March 1, 1997)
2. Wang, X.Z., Hong, J.R.: Learning Algorithm of Decision Tree Generation for Interval-Valued Attributes. *Journal of Software* **9**(8), 637–640 (1998)
3. Quinlan, J.R.: Induction of Decision Tree. *Machine Learning* **1**(1), 81–106 (1986)
4. Wu, X.D., Kumar, V., Quinlan, J.R., et al.: Top 10 algorithms in data mining. *Knowledge and Information Systems* **14**(1), 1–37 (2008)
5. Breiman, L., Friedman, J.H., Olshen, R.A., et al.: *Classification and Regression Tree*. Wadsworth International Group (1984)
6. Ben-David, A., Sterling, L., Pao, Y.H.: Learning and classification of monotonic ordinal concepts. *Computational Intelligence* **5**(1), 45–49 (1989)
7. Zopounidis, C., Doumpos, M.: Multicriteria classification and sorting methods-A literature review. *European Journal of Operational Research* **138**, 229–246 (2002)
8. Krzysztof, D., Wojciech, K., Roman, S.: Ensemble of Decision Rules for Ordinal Classification with Monotonicity Constraints
9. Potharst, R., Bioch, J.: Decision trees for ordinal Classification. *Intelligent Data Analysis* **4**(2), 97–112 (2000)
10. Cao-Van, K., Baets, B.D.: Growing decision trees in an ordinal setting. *International Journal of Intelligent Systems* **18**, 733–750 (2003)
11. Baril, N., Feelders, A.J.: Nonparametric Monotone Classification with MOCA/ICDM, pp. 731–736 (2008)
12. Potharst, R., Feelders, A.J.: Classification trees for problems with monotonicity constrains. *SIGKDD Explorations* **4**(1), 1–10 (2002)

13. Potharst, R., Bioch, J.C.: Decision trees for ordinal classification. *Intelligent Data Analysis* **4**, 97–111 (2000b)
14. Kotlowski, W., Slowinski, R.: Rule learning with monotonicity constrains. In: *Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada*, pp. 537–544 (2009)
15. Hu, Q.H., Guo, M.Z., Yu, D.R., et al.: Information entropy for ordinal classification. *Science China Information Sci.* **53**(6), 1188–1200 (2010)
16. Hu, Q., Che, X., et al.: Rank Entropy-Based Decision Trees for Monotonic Classification. *IEEE Transactions on Knowledge and Data Engineering* **24**(11), 2052–2064 (2012)

Extended Space Decision Tree

Md. Nasim Adnan¹(✉), Md. Zahidul Islam¹, and Paul W.H. Kwan²

¹ Centre for Research in Complex Systems (CRiCS), School of Computing and Mathematics,
Charles Sturt University, Bathurst, NSW 2795, Australia
{madnan, zislam}@csu.edu.au

² School of Science and Technology, University of New England, Armidale,
NSW 2351, Australia
paul.kwan@une.edu.au

Abstract. An extension of the attribute space of a dataset typically increases the prediction accuracy of a decision tree built for this dataset. Often attribute space is extended by randomly combining two or more attributes. In this paper, we propose a novel approach for the space extension where we only choose the combined attributes that have high classification capacity. We expect the inclusion of these attributes in the attribute space increases the prediction capacity of the trees built from the datasets with the extended space. We conduct experiments on five datasets coming from the UCI machine learning repository. Our experimental results indicate that the proposed space extension leads to the tree of higher accuracy than the case where original attribute space is used. Moreover, the experimental results demonstrate a clear superiority of the proposed technique over an existing space extension technique.

Keywords: Extended space decision forest · Decision tree · Prediction accuracy

1 Introduction

Nowadays the amount of data is increasing in an unprecedented pace — so much that 90% of the data in the world today have been generated in the last two years alone [1]. The huge volume of data makes it almost impossible for the domain experts to infer any useful knowledge or pattern manually. So the need for automatic means for transforming data into useful information arises. Data mining is about automatically discovering useful information from large datasets [2]. Classification and clustering are two widely used data mining tasks that are applied for knowledge discovery and pattern recognition.

Classification aims to generate a function that maps the set of classifier attributes $\{A_1, A_2, \dots, A_m\}$ to a predefined class attribute C [2]. The function is commonly known as the classification model which is generally used for two main purposes: 1. knowledge discovery, and 2. the prediction of the class value of an unlabeled record. An unlabeled record is a record that has no class value assigned. Records representing the patients that are not diagnosed with any disease are the example of unlabeled records in a hospital dataset where Diagnosis is the class attribute.

There are different types of classifiers including Decision Trees [3, 4], Bayesian Classifiers [5, 6], Artificial Neural Networks [7, 8, 9], and Support Vector Machines

[10]. Among these classifiers, decision trees are very popular as they can be easily broken down to generate logic rules, which help us to infer valuable knowledge [11]. Due to their high popularity, decision trees with better prediction accuracy can render huge impact on many sensitive application areas such as medical diagnosis.

There are many decision tree building algorithms such as CART [3], ID3 [4] and C4.5 [4], [12] that aim to achieve high prediction accuracy for the generated decision trees. Typically, they use different strategies (such as different splitting measures) to build the trees while one of the main aims is to achieve higher prediction accuracy. In this paper we take a different path to increase the prediction accuracy. Inspired by extended space decision forests, we propose a novel approach for generating new attributes with good classification capacity from the original non-class attributes of a training dataset that is a dataset where all records are labeled with class values. The newly generated attributes are then added in the training dataset along with the original attributes in order to build a new training dataset with extended number of attributes. The new training dataset with the extended attribute is also commonly known as the extended space dataset. The extended space dataset is then used to generate a decision tree using an algorithm such as the C4.5 algorithm. Due to the inclusion of new attributes with high classification capacity in the extended space dataset, we expect to build a decision tree with high prediction accuracy when a decision tree algorithm is applied on the extended space dataset.

The paper is organized as follows: In Section 2, we give an overview of the C4.5 algorithm. Then, Section 3 presents several decision forest building algorithms that are commonly used in the literature. In Section 4 we introduce our technique for extended space decision tree. Then, Section 5 presents the experimental results in detail. Finally, Section 6 presents the concluding remarks.

2 C4.5 Decision Tree Algorithm

Since our proposed technique uses a decision tree algorithm such as C4.5, in this study we provide a brief introduction to the algorithm.

C4.5 algorithm recursively builds a decision tree from a training dataset. The induction process starts by computing the Gain Ratio [4], [12] (splitting measure of C4.5 algorithm) of all non-class attributes. Then the attribute having the highest Gain Ratio value is selected as a splitting attribute to divide the training dataset D into a set of mutually exclusive horizontal segments/partitions [4], [12, 13, 14]. The purpose of this splitting is to create a purer distribution of class values in the succeeding segments/partitions than the distribution in D .

The selection of the best splitting attribute continues in a recursive manner within each subsequent data segment D_i until either of the following two conditions is met: 1. every partition produces the purest class distribution, 2. another stopping criterion is satisfied. “Purest class distribution” means the presence of a single class value for all records.

It is worth mentioning that decision tree is very sensitive to the training dataset [2]. That is, if the training dataset is slightly perturbed by adding or removing some records or attributes, the resultant decision tree can be very different from that generated from

the unperturbed training dataset. In the literature, this property of decision trees has been effectively exploited to explore the horizon of decision forest [15].

3 Decision Forests

Decision forest is an ensemble of decision trees where an individual decision tree acts as a base classifier that is used to classify the records. The overall classification of the ensemble is performed by taking a vote based on the individual classification made by each decision tree [2]. There are many different approaches to generate decision forests. In order to achieve better ensemble accuracy, a decision forest needs both accurate and diverse individual decision trees [15, 16]. An accurate individual decision tree can be obtained by feeding a dataset into a decision tree building algorithm such as C4.5. However, if all the individual decision trees generate similar classification results then there is no purpose of constructing a decision forest. There are many decision forest building algorithms which intend to diversify decision trees by modifying a training dataset, since decision trees are typically known to be sensitive to a training dataset. Different forest algorithms modify training datasets in different ways. Since we present a novel approach to modify a training dataset, we now summarize some relevant forest building algorithms as follows.

Bagging [17]: Bagging generates new training dataset D_i iteratively ($i = 1, 2, \dots, k$) where the records of D_i are randomly chosen from the original dataset D in such a way that a single record can be chosen multiple times. This approach of generating a new training dataset is known as bootstrap sampling [18]. A decision tree building algorithm is then applied on each training dataset D_i ($i = 1, 2, \dots, k$) in order to build k number of trees for the forest.

Boosting [19, 20]: In Boosting, a decision tree is obtained from a weighted training dataset D_i where each record of D_i is assigned a weight. While building the first tree T_1 of the forest each record of the training dataset D_1 is assigned an equal weight. The records of D_1 are then classified using T_1 . The records that are misclassified are then assigned higher weights in the new training dataset D_2 where a new tree T_2 is built from. The process of assigning weights and building trees continues for a user defined k number of iterations.

Random Subspace [16]: In building a decision tree, Random Subspace algorithm considers a subset (instead of an extended space) of the original attribute space at each node splitting event of a decision tree. Random Subspace uses all the records of the training dataset in building every decision tree.

Random Forest [21]: Random Forest is a combination of the Bagging and Random Subspace algorithms. In Random Forest, the Random Subspace algorithm is applied on bootstrap samples of a training dataset.

Rotation Forest [22]: In Rotation Forest, at first a new training dataset is obtained by applying Principal Component Analysis (PCA) on the original training dataset. Then algorithms such as Bagging, Random Subspace and Random Forest are applied to the new training dataset to generate the decision forest.

4 Our Technique

4.1 Basic Concept

From the decision forest building algorithms discussed above, we notice different techniques of manipulating the training dataset. For example, in Random Subspace algorithm [16], the author also suggested generating new attributes if the number of original attributes in the training dataset is too small to use the Random Subspace algorithm effectively. Recently, M. F. Amasyali and O. K. Ersoy [23] demonstrated that decision forest building algorithms such as Bagging, Random Subspace, Random Forest, and Rotation Forest perform better when applied on extended attribute space. The authors [23] suggested extending the attribute space in the following way.

Let, D be the training dataset with d original attributes. Thus, the original attribute space $A_O = \{A_1, A_2, \dots, A_d\}$. Then any attribute pair (say A_i and A_j) is randomly selected from A_O and combined using difference operator (by subtracting the values of A_i and A_j) to form a new attribute $= A_i - A_j$, where the index i is less than the index j . This process iterates d times and thus d number of new attributes, $A_N = \{A'_1, A'_2, \dots, A'_d\}$, are generated. Finally, the newly generated attribute space is added to the original attribute space to form the extended attribute space $A_E = A_O \cup A_N$.

M.F. Amasyali and O. K. Ersoy [23] presented detailed experimental results on the effect of attribute space extension for Bagging, Random Subspace, Random Forest and Rotation Forest. It appears that all these decision forest building algorithms deliver better ensemble accuracy when applied on the extended attribute space than the original attribute space. The classification/prediction accuracy of each individual tree was measured [23] in order to compute the Average Individual Accuracy (AIA) of a decision forest. It appears that various forest building algorithms such as Bagging, Random Subspace and Random Forest achieve higher AIA when they use an extended space training dataset compared to the original attribute space for the dataset.

Therefore, it is evident that the extension of an original attribute space typically results in higher accuracy of a decision tree built from the extended space dataset. The reason behind the higher accuracy could be the simple fact that due to the generation of new attributes combining two attributes (or any other number of attributes) one often gets a new attribute with better classification capacity. This phenomenon is similar to drug-drug interaction [24] in which two drugs may have a better result when both are administered together than when they are used individually on a patient.

While the advantage of the space extension is apparent, the possible technique with the best outcome remains unclear. First, it is not always possible for a data miner to add a completely new attribute (such as the blood sugar level of the patients) in a dataset, even if the new attribute is suspected to have a high classification capacity, if sugar levels of the patients were not tested before. Second, if two attributes are combined randomly as it is done in the literature [23], the chance of getting a new attribute with high classification capacity is not as high as it would be in the case where we combine the attributes systematically (not randomly) in order to find the set of new attributes with high classification capacity.

Therefore, in this paper we present a novel approach for attribute space extension where we aim to systematically generate new attributes with high classification capacity. We expect that the addition of the new attributes with high classification capacity should generally increase the accuracy of a decision tree generated from the extended dataset.

4.2 Extended Space Decision Tree

We first present the basic steps of our proposed technique as follows.

Step 1: Combine existing attributes to produce a set of candidate attributes.

Step 2: Select new attributes from the set of candidate attributes.

Step 3: Build a decision tree from the extended space dataset.

We now introduce the basic steps as follows.

Step 1: Combine existing attributes to produce a set of candidate attributes.

In our technique, we extend the original attribute space by generating new attributes with high classification capacity. The new attributes are generated by combining k number of existing attributes (see Step 1 of Algorithm 1). Therefore, we get a set of candidate attributes A_C , where the size of the set is $|A_C| = {}^d C_k$. Note that our technique is not restricted to any specific value of k . The common practice in the literature is to take $k = 2$ [23], [25]. In the experimental section of this study we use two different k values, namely 2 and 3.

In this study, we consider datasets having only categorical attributes for the following reasons. The obvious process of combining two or more categorical attributes is concatenation. On the other hand, for numerical attributes there are many different possible approaches for concatenation such as addition, division, and multiplication [23]. In this study we focus on the impact of space extension and do not focus on different approaches of combining the numerical attributes. Therefore, we choose to use the datasets that have only categorical attributes so that we can neutralize the impact of different ways of combining attributes instead focusing on the impact of space extension.

However, for the sake of the completeness of our technique we propose to first categorize/discretize [26, 27] all numerical attributes and then use them as any other categorical attributes. This also solves the problem when we have a numerical and a categorical attribute that need to be combined. Nevertheless, discretization is also not the focus of this study and we aim to work on this in future.

Step 2: Select new attributes from the set of candidate attributes.

We now compute the Gain Ratio [4], [12] of each of the ${}^d C_k$ number of candidate attributes that are obtained from Step 1. Based on the Gain Ratio of the candidate attributes we select the set of the best d' attributes ($A_{d'}$) with the highest Gain Ratio values. The selected d' attributes are then added in the original dataset. Therefore, we get the extended attribute space $A_E = A_O \cup A_{d'}$ (see Step 2 of Algorithm 1).

The Gain Ratio of an attribute indicates the classification capacity of the attribute. Therefore, the best d' attributes based on the Gain Ratio represent the set of d' attributes with the best classification capacity among the ${}^d C_k$ candidate attributes.

Our proposed technique is not restricted to any specific d' value. In the literature different d' values (such as d , $2d$, and $3d$) have been tested [23]. In Section 5 we also experiment on two different d' values such as $d/2$ and d .

Step 3: Build a decision tree from the extended space dataset.

In this step we apply any existing decision tree algorithm such as C4.5 on the extended space training dataset obtained from Step 2. We therefore build a decision tree that tests the attributes from the set of extended space attributes A_E (see Step 3 of Algorithm 1).

This decision tree can be used for classification on the training dataset and prediction on the testing dataset that contains unlabeled records only. Note that the attributes that have been combined in the training dataset need to be combined in the testing dataset as well in order to allow us to use the decision tree for the prediction purpose.

5 Experimental Results

We conduct an elaborative experimentation on five datasets that are publicly available on the UCI Machine Learning Repository [28]. The datasets used in the experimentation are listed in Table 1. All attributes for every dataset shown in Table 2 are categorical.

All the results presented in this paper are obtained using 10-fold-cross-validation [14] (10-CV) for every dataset. All the prediction accuracies reported in this paper are in percentage. The best results are presented in bold-face.

The main objectives of our experimentation are as follows. First, we explore the effectiveness of the space extension in building decision trees that have higher prediction accuracy. Second, we empirically investigate the suitability of the size (d') of space extension. Third, we examine the impact of different values of k .

We use $k = 2$ and $k = 3$. For $k = 2$, new attributes are generated by concatenating two different existing attributes. Therefore, we get a set of candidate attributes $A_{C_2} = \{A_1A_2, A_1A_3, \dots, A_1A_d, A_2A_3, A_2A_4, \dots, A_2A_d, \dots, A_{d-1}A_d\}$ with ${}^d C_2$ number of new attributes. Similarly, for $k = 3$ we get $A_{C_3} = \{A_1A_2A_3, A_1A_2A_4, \dots, A_1A_{d-1}A_d, \dots, A_{d-2}A_{d-1}A_d\}$ with ${}^d C_3$ number of new attributes.

We also use two different d' values: $d/2$ and d as the attribute extension parameter. We first select the best $d/2$ attributes ($A_{C_2}^{d/2}$) from the set of the A_{C_2} candidate attributes in order to form the extended attribute space $A_{E,C_2}^{d/2} = A_O \cup A_{C_2}^{d/2}$. Similarly, we select the best d attributes ($A_{C_2}^d$) to generate $A_{E,C_2}^d = A_O \cup A_{C_2}^d$. We also select the best $d/2$ attributes ($A_{C_3}^{d/2}$) from the A_{C_3} candidate attributes in order to form the extended attribute space $A_{E,C_3}^{d/2} = A_O \cup A_{C_3}^{d/2}$. Finally, the best d attributes ($A_{C_3}^d$) are selected to generate the extended attribute space $A_{E,C_3}^d = A_O \cup A_{C_3}^d$.

Algorithm 1: Extended Space Decision Tree

Input: Training Dataset D with original attribute space $A_O = \{A_1, A_2, \dots, A_d\}$, number of attributes to be combined k , size of extension d'

Output: A Decision Tree (T).

Step 1

$A_C = \text{produce_candidate_attributes}(D, A_O, k);$ /* A_C is the set
of all candidate attributes */

End Step 1**Step 2**

$G_C = \text{compute_Gain_Ratio}(A_C, D);$
 $A_{C'} = \text{Sort_and_Match}(G_C, A_C);$ /* $A_{C'}$ is the sorted
set of all candidate attributes */
 $A_{d'} = \text{Pick_Best}(d', A_{C'});$
 $A_E = \text{Get_Extended_Space}(A_O, A_{d'});$ /* $A_E = A_O \cup A_{d'}$ */
 $D' = \text{Extend_Dataset}(D, A_E);$

End Step 2**Step 3**

$T = \text{build_tree}(D')$ /* Build a decision tree using an
existing algorithm such as the C4.5 Algorithm */
 return T ;

End Step 3**Table 1.** Datasets used in the experiment

Dataset Name	Number of Attributes	Number of Records	Domain Size of the Class Attribute
Car Evaluation	6	1728	4
Tic-Tac-Toe	9	958	2
Balance Scale	4	625	3
Soybean	35	47	4
Lenses	4	24	3

In Table 2, we present the prediction accuracy of C4.5 decision trees built from datasets having the original attribute space A_O and the extended attribute space $A_{E,C_2}^{d/2}$. In four out of five datasets the trees built from $A_{E,C_2}^{d/2}$ have higher prediction accuracy than the trees built from A_O . Therefore, it indicates that our proposed space extension technique can produce decision trees with higher prediction accuracy.

Table 2. Prediction accuracies for AO and $A_{E,C_2}^{d/2}$

Dataset Name	C4.5 on A_O	C4.5 on $A_{E,C_2}^{d/2}$
Car Evaluation	94.0950	93.9250
Tic-Tac-Toe	82.5850	83.5170
Balance Scale	65.6000	70.4490
Soybean	72.2730	72.9550
Lenses	83.3330	86.6670
Average	79.5772	81.5026

In Table 3, we explore the impact of the size of the space extension where we extend the space by adding d attributes instead of $d/2$ attributes. The results in Table 2 and Table 3 indicate that for these datasets the extension of $d/2$ attributes are more suitable than d attributes.

Table 3. Prediction accuracies for AO and A_{E,C_2}^d

Dataset Name	C4.5 on A_O	C4.5 on A_{E,C_2}^d
Car Evaluation	94.0950	94.4430
Tic-Tac-Toe	82.5850	77.9200
Balance Scale	65.6000	66.2380
Soybean	72.2730	61.3640
Lenses	83.3330	83.3330
Average	79.5772	76.6596

In Table 4 and Table 5 we examine the impact of $k = 3$ compared to $k = 2$ that can be studied from Table 2 and Table 3. We observe that combining three (3) attributes at a time ($k = 3$) disturbs the improvement of the space extension. For example, comparing Table 2 and Table 4 we find that the trees built from $A_{E,C_2}^{d/2}$ achieve higher accuracy. Similar observation can be made from Table 3 and Table 5. Out of these four tables (from Table 2 to Table 5), we achieve the highest accuracy in Table 2 and the lowest in Table 5.

The complexity of C4.5 is $O(nm^2)$ where n is the number of records and m is the number of attributes [29]. Therefore, the d attribute extension introduces higher complexity than the $d/2$ extension. Moreover, from the four tables we also observe that the $d/2$ space extension makes a better improvement than the d extension. This is also shown clearly in Fig. 1, where the combined average values of $k = 2$ and $k = 3$ are shown for both $(d+d)$ and $(d+d/2)$.

Table 4. Prediction accuracies for AO and $A_{E,C_3}^{d/2}$

Dataset Name	C4.5 on A_o	C4.5 on $A_{E,C_3}^{d/2}$
Car Evaluation	94.0950	94.7340
Tic-Tac-Toe	82.5850	74.6340
Balance Scale	65.6000	68.9270
Soybean	72.2730	73.6360
Lenses	83.3330	88.3330
Average	79.5772	80.0528

Table 5. Prediction accuracies for AO and A_{E,C_3}^d

Dataset Name	C4.5 on A_o	C4.5 on A_{E,C_3}^d
Car Evaluation	94.0950	77.8410
Tic-Tac-Toe	82.5850	73.9140
Balance Scale	65.6000	64.6310
Soybean	72.2730	61.3640
Lenses	83.3330	80.0000
Average	79.5772	71.5500

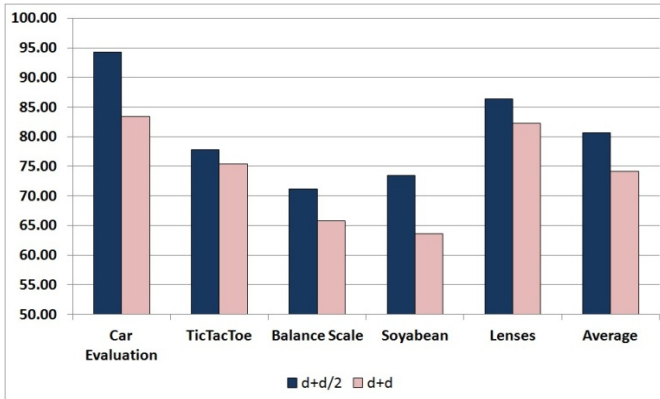


Fig. 1. Comparison between $d/2$ and d

Therefore, these initial experiments indicate a clear improvement of accuracy based on our space extension technique. The improvement appears to be prominent when we use $k = 2$ and $d' = d/2$.

We further investigate the effectiveness of our proposed technique by comparing it with an existing space extension technique where two attributes are randomly chosen for producing a new combined attribute [23]. We implement the technique to extend the

attribute space by d as originally suggested [23] in order to produce the new attribute space $A_{E,C_2}^{nd d}$. The results presented in Table 6 indicate a clear superiority of our proposed technique over the existing one.

Table 6. Prediction accuracies for $A_{E,C_2}^{d/2}$ and $A_{E,C_2}^{nd d}$

Dataset Name	C4.5 on $A_{E,C_2}^{d/2}$	C4.5 on $A_{E,C_2}^{nd d}$
Car Evaluation	93.9250	93.3020
Tic-Tac-Toe	83.5170	78.4880
Balance Scale	70.4490	67.6350
Soybean	72.9550	62.9550
Lenses	86.6670	80.0000
Average	81.5026	76.4760

In Table 7, we present the average depth of the decision trees. The trees built from A_{E,C_3}^d have the lowest depth on an average whereas the trees built from the original datasets have the highest depth. Among the trees built from the extended space datasets, the trees built from $A_{E,C_2}^{d/2}$ have the highest depth. From the previous tables we also find that the trees built from $A_{E,C_2}^{d/2}$ have the highest accuracy while the trees built from A_{E,C_3}^d have the lowest accuracy.

Table 7. Depth of decision trees

Dataset Name	C4.5 on A_o	Selection Space: A_{C_2}			Selection Space: A_{C_3}	
		C4.5 on $A_{E,C_2}^{d/2}$	C4.5 on A_{E,C_2}^d	C4.5 on $A_{E,C_2}^{nd d}$	C4.5 on $A_{E,C_3}^{d/2}$	C4.5 on A_{E,C_3}^d
Car Evaluation	6.0	5.3	5.2	6.0	6.0	4.2
Tic-Tac-Toe	6.9	7.0	6.0	5.3	2.6	2.1
Balance Scale	4.0	3.0	3.0	3.0	2.0	2.0
Soybean	2.4	2.5	2.3	2.4	2.8	2.1
Lenses	4.0	4.0	3.9	4.0	4.0	3.8
Average	4.7	4.4	4.1	4.1	3.5	2.8

In the datasets having A_{E,C_3}^d we use $k = 3$ and $d' = d$. For $k = 3$ we get new multivariate attributes combining three original attributes. We know that the multivariate attributes have higher representation power than original univariate attributes [28]. Therefore, if a tree uses a multivariate attribute in a node the tree is likely to have a smaller depth. Moreover, when attribute space is extended with d newly generated multivariate attributes there is a higher probability of having more multivariate nodes in the resultant

tree than when attribute space is extended with $d/2$ newly generated multivariate attributes. Hence, it makes sense to have the shallowest trees when they are built from A_{E,C_3}^d .

6 Conclusion

Decision trees are useful in knowledge discovery and future prediction. Therefore, it is important to build decision trees with higher prediction accuracy. Accuracy of the trees generally increases when they are built from datasets having extended attribute space. In this paper we propose a novel technique that extends the attribute space systematically (instead of randomly) aiming to include new attributes with high classification capacity. Therefore, the proposed technique combines the existing attributes and then picks only those having high Gain Ratio since it indicates the classification capacity of an attribute.

Our initial experiments indicate that the proposed space extension achieves higher accuracy than the case where original attribute space is used. Moreover, the experimental results demonstrate a clear superiority of the proposed technique over an existing technique.

We also observe the best improvement for the $d/2$ extension with $k = 2$. However, this is an early result based on a relatively small number of datasets. Therefore, our future plan is to carry out extensive experiments on more datasets in order to explore the impact and reasons for the best setting.

References

1. IBM Co., Bringing big data to the Enterprise. <http://www-01.ibm.com/software/au/data/bigdata/> (last accessed: July 25, 2013)
2. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education Inc., Boston (2006)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International Group, U.S.A (1985)
4. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag New York Inc., NY (2008)
6. Mitchell, T.M.: Machine Learning. McGraw-Hill, U.S.A (1997)
7. Jain, A.K., Mao, J.: Artificial Neural Network: A Tutorial. Computer **29**(3), 31–44 (1996)
8. Zhang, G.P.: Neural Networks for Classification: A Survey. IEEE Transactions on Systems, Man, and Cybernetics **30**, 451–462 (2000)
9. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting **14**, 35–62 (1998)
10. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery **2**, 121–167 (1998)
11. Murthy, S.K.: Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery **2**, 345–389 (1998)

12. Quinlan, J.R.: Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research* **4**, 77–90 (1996)
13. Islam, M.Z.: EXPLORE: A Novel Decision Tree Classification Algorithm. In: MacKinnon, L.M. (ed.) *BNCOD 2010. LNCS*, vol. 6121, pp. 55–71. Springer, Heidelberg (2012)
14. Islam, M.Z., Giggins, H.: Knowledge Discovery through SysFor – a Systematically Developed Forest of Multiple Decision Trees. In: *Proceedings of the 9th Australian Data Mining Conference*, Ballarat, Australia, pp. 195–204 (2011)
15. Polikar, R.: Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine* **6**, 21–45 (2006)
16. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **20**(1), 832–844 (1998)
17. Breiman, L.: Bagging predictors. *Machine Learning* **24**(2), 123–140 (1996)
18. Han, J., Kamber, M.: *Data Mining Concepts and Techniques*. Morgan Kaufmann, San Francisco (2006)
19. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: *Proceedings of the Thirteenth International Conference on Machine Learning*, July 3-6, Bari, Italy, pp. 148–156 (1996)
20. 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)
21. Breiman, L.: Random Forests. *Machine Learning* **45**(1), 5–32 (2001)
22. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation Forest: A New Classifier Ensemble Method. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **28**(2), 1619–1630 (2006)
23. Amasyali, M.F., Ersoy, O.K.: Classifier Ensembles with the Extended Space Forest. *IEEE Transaction on Knowledge and Data Engineering* **26**, 549–562 (2014)
24. NPS MEDICINEWISE, Drug Interaction. <http://www.nps.org.au/media-centre/media-releases> (last accessed: February 21, 2014)
25. Ahmed, A., Brown, G.: Random Projection Random Discretization Ensembles – Ensembles of Linear Multivariate Decision Trees. *IEEE Transaction on Knowledge and Data Engineering* (August 2013) ISSN: 1041–4347 (to be published)
26. Kurgan, L.A., Cios, K.J.: CAIM Discretization Algorithm. *IEEE Transaction on Knowledge and Data Engineering* **16**, 145–153 (2004)
27. Kotsiantis, S., Kanellopoulos, D.: Discretization Techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering* **32**, 47–58 (2006)
28. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/datasets.html> (last accessed: December 15, 2013)
29. Su, J., Zhang, H.: A Fast Decision Tree Learning Algorithm. In: *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*, pp. 500–505, Boston, U.S.A., July 16-20, (2006)

Monotonic Decision Tree for Interval Valued Data

Hong Zhu, Junhai Zhai^(✉), Shanshan Wang, and Xizhao Wang

Key Laboratory of Machine Learning and Computational Intelligence,
College of Mathematics and Computer Science, Hebei University,
Baoding 071002, China
mzczjh@126.com

Abstract. Traditional decision tree algorithms for interval valued data only can deal with non-ordinal classification problems. In this paper, we presented an algorithm to solve the ordinal classification problems, where both the condition attributes with interval values and the decision attributes meet the monotonic requirement. The algorithm uses the rank mutual information to select extended attributes, which guarantees that the outputted decision tree is monotonic. The proposed algorithm is illustrated by a numerical example, and a monotonically consistent decision tree is generated. The design of algorithm can provide some useful guidelines for extending real-valued to interval-valued attributes in ordinal decision tree induction.

Keywords: Interval valued data monotonic classification · Decision tree · Rank mutual information

1 Introduction

Decision tree [1] is a well known inductive learning algorithm which has been widely applied to data mining [2], pattern recognition [3], decision making [4] etc. Decision tree learning algorithms typically use a heuristic to select the extended attributes. A number of heuristics can be found in references but the information entropy introduced by Quinlan [1] in ID3 algorithm is considered as a popular one. In comparison with the other decision tree algorithms, ID3 algorithm is more effective in dealing with noise data and missing data. Further more, Quinlan extended the ID3 algorithm to the C4.5 version [5] which uses information gain ratio as the heuristic to select the extended attributes and has been regarded as one of the most popular decision tree procedures. C4.5 overcomes the disadvantage of ID3 algorithm that tends to choose attributes with more values and therefore obtains a better performance in comparison with ID3. Based on Quinlan's seminal work of decision tree induction, many improved versions of ID3 have been proposed by different researchers, for example, Cheng, et al. improved the ID3 by revising the representation of relation between conditional attributes and decision attribute [6]. Noting that the potential uncertainty in the process

of classification cannot be expressed explicitly and a small change of attribute value can lead to radical changes of classification results, a probabilistic method to construct the decision tree was presented by Quinlan [7]. Following this work, Hong et al. in [8] analyzed the optimization principle of decision tree inductive learning from the point of view of the example learning optimization and proposed a new method to select the extended attributes, and furthermore, Fayyad et al. in [9] extended the probabilistic decision tree learning algorithm to the case of continuous-valued attributes. Incorporating the uncertainty including the fuzziness and ambiguity into the process of decision tree generation, Yuan et al. in [10] proposed a scheme of fuzzy decision tree induction.

The above-mentioned algorithms can only solve general classification problems where we do not require that the decision attributes are monotonic with the conditional attributes. Practically we often meet with another kind of classification problems named ordinal or monotonic classification in which the condition attributes and the decision attribute are well-ordered but the classification rules extracted from decision trees are required to be order-preserving [11]. Ordinal classification problems are ubiquitous in real world and it can be found in many areas such as credit quality rating, scholarship evaluation, and bankruptcy risk evaluation etc. The ordinal classification problems have been investigated by many researchers. For example, an algorithm of order-preserving tree-generation and an algorithm for repairing non-monotonic decision trees were proposed in [11] to handle multi-attribute classification problems with k linearly ordered classes; and a simple method that enables standard classification algorithms to make use of ordering information in class attributes was presented in [12] by transforming k -class ordinal problems to $k-1$ binary class problems. Another example is that Xia et al. [17] extended the Gini impurity used in CART to ordinal classification and obtained a new splitting rule for generation of decision trees with continuous attributes.

The ordinal classification problem can be investigated from the viewpoint of rough sets. For example, Greco et al. [13] used dominance rough set which is an extension of classical rough sets theory initiated by Pawlak [14] to solve monotonic classification problem. Although dominance rough set provides a formal theoretic framework for studying the monotonic classification problems [15, 16], but it may not be effective in practice due to its sensitivity to noisy and missing data.

Hu et al. in 2010 gave the definition of rank entropy and rank mutual information [18], which combines the advantage of information entropy and dominance rough sets. Hu et al. analytically and experimentally confirm that the rank mutual information not only can measure the monotonous consistency in monotonic classification but also can produce a robust algorithm handling noisy samples. Furthermore in [19], Hu et al. developed an algorithm based on the rank mutual information for generating monotonic decision trees.

In most algorithms of decision tree learning, the attributes can be classified into two categories according to their values: one is called nominal attribute while the other is called numerical attribute. For nominal attribute values the domain is a finite set without order, but for numerical attributes, it is a well-ordered set.

Wang X Z et al. noted in [20] that interval-valued attribute is semi-ordered, located between the well-order and non-order. Then an interval-valued algorithm for generating decision trees was developed in [20]. The algorithm, which selects extended attributes by minimizing segment information entropy, can be regarded as an extended version of ID3 from real-valued attributes to interval-valued attributes. This algorithm was developed for non-ordered classification problem and could not solve classification problems in which both conditional attributes and decision attributes are ordered or semi-ordered. In this paper, we extend the work in [20] and propose an algorithm which can deal monotonic classification problems with interval-valued attributes. Our proposed algorithm, which selects extended attributes by minimizing rank mutual information to generate a decision tree, can give an order-preserving prediction result for unseen samples with interval values.

The paper is organized as follows. Section 2 provides preliminaries about monotonic classification problem for interval valued samples. Section 3 develops our new algorithm for generating decision trees with interval-valued attributes. Section 4 lists our experimental results and Section 5 concludes this paper.

2 Monotonic Classification Problem for Interval Valued Samples

In this section, we briefly review the basic concepts used in this paper.

2.1 Interval Value

Definition 1. Let $a^L, a^U \in R$, and $a^L \leq a^U$, $\tilde{a} = [a^L, a^U] = \{x | a^L \leq x \leq a^U\}$, \tilde{a} is called an interval value, where, a^L is the left endpoint of \tilde{a} , a^U is the right endpoint of \tilde{a} . Specially, if $a^L = a^U$, \tilde{a} is a real number.

Definition 2. When \tilde{a} and \tilde{b} are both interval value, let $\tilde{a} = [a^L, a^U]$, $\tilde{b} = [b^L, b^U]$, $l_a = a^U - a^L$, $l_b = b^U - b^L$, then $p(\tilde{a} \geq \tilde{b})$ is called the possibility degree of $\tilde{a} \geq \tilde{b}$, where

$$p(\tilde{a} \geq \tilde{b}) = \frac{\min\{l_{\tilde{a}} + \bar{l}_{\tilde{b}}, \max(a^U - b^L, 0)\}}{l_{\tilde{a}} + l_{\tilde{b}}} \tag{1}$$

Definition 3. If $p(\tilde{a} \geq \tilde{b}) \geq 0.5$, then \tilde{a} is considered better than \tilde{b} , i.e. $\tilde{a} \geq \tilde{b}$; otherwise, \tilde{a} is considered worse than \tilde{b} , i.e. $\tilde{a} \leq \tilde{b}$.

2.2 Monotonic Classification Problem

Let $DT = (U, A, D)$ be a decision table, where $U = \{x_1, \dots, x_n\}$ is a set of objects and A is a set of attributes; D is a finite ordinal set of decisions. The value of x_i in attributes $a \in A$ or D is denoted by $v(x_i, a)$ or $v(x_i, D)$ respectively. We say x_j is not worse than x_i in terms of a or D if $v(x_i, a) \leq v(x_j, a)$ or $v(x_i, D) \leq v(x_j, D)$,

denoted this relation by $x_i \leq_a x_j$ and $x_i \leq_D x_j$ respectively. Correspondingly, we can also define $x_i \geq_a x_j$ and $x_i \geq_D x_j$. Given $B \subseteq A$, we say $x_i \leq_B x_j$ if $v(x_i, a) \leq v(x_j, a)$ for $\forall a \in B$ [19].

A monotonically ordinal classification function should satisfy the following constraints:

$$x_i \leq x_j \rightarrow f(x_i) \leq f(x_j), \forall x_i, x_j \in U.$$

Definition 4. Let $DT = (U, A, D)$ be a decision table, $B \subseteq A$. We say DT is B -monotonically consistent if $\forall x_i, x_j \in U, x_i \leq_B x_j$, we have $x_i \leq_D x_j$.

Definition 5. Let DT be B -monotonically consistent, for $\forall x \in U, B \subseteq A, a \in B$, we associate x with the following sets:

$$[x]_a^{\geq} = \{y \in U : y \geq_a x\}; \tag{2}$$

$$[x]_B^{\geq} = \{y \in U : y \geq_B x\}; \tag{3}$$

$$[x]_a^{\leq} = \{y \in U : y \leq_a x\}; \tag{4}$$

$$[x]_B^{\leq} = \{y \in U : y \leq_B x\}. \tag{5}$$

2.3 Rank Mutual Information

It is well known that the measure of mutual information derived from Shannon entropy outperforms the measures of Gini and dependency in decision tree construction [22–24]. In addition, mutual information also performs well in feature selection for evaluating quality of features [25], discretization for evaluating cutting sets [26], and registration of images [27], etc. A lot of references show that the mutual information can widely be used to measure the correlation between random variables in classification and regression problems.

Shannon entropy cannot reflect the monotone consistency between condition attributes and decision attribute in monotonic classification problems. Although, dominance rough set gives us a formal framework for analyzing consistency in monotonic classification task, it is heavily sensitive to noisy samples such as missing data and data perturbation. Several mislabeled samples might completely change the trained decision models, which seriously downgrades the application capability of dominance rough sets. In view of these reasons, the concept of ordinal mutual information was proposed in [18].

Definition 6. Given $DT = (U, A, D)$, $B \subseteq A, C \subseteq A$. The ascending rank mutual information (ARMI) of the set U between B and C is defined as

$$RMI^{\leq}(B, C) = -\frac{1}{|U|} \sum_{i=1}^n \log \frac{|[x_i]_B^{\leq}| \times |[x_i]_C^{\leq}|}{|U| \times |[x_i]_B^{\leq} \cap [x_i]_C^{\leq}|} \tag{6}$$

and the descending rank mutual information (DRMI) of the set U regarding B and C is defined as

$$RMI^{\geq}(B, C) = -\frac{1}{|U|} \sum_{i=1}^n \log \frac{|[x_i]_{\bar{B}}^{\geq}| \times |[x_i]_{\bar{C}}^{\geq}|}{|U| \times |[x_i]_{\bar{B}}^{\geq} \cap [x_i]_{\bar{C}}^{\geq}|} \tag{7}$$

The effect of mutual information is to measure the monotonic consistency between attribute sets B and C . Different from Spearmans rank correlation coefficient [28], RMI can be used to compute the relevance between two sets of variables, instead of two variables. In monotonic classification problems, monotonic consistency is crucial factor for inductive learning. This factor which has not an important impact on the entropy based heuristic information during the process of generating a decision tree is not considered in traditional classification problems.

The rank mutual information (RMI) of the set U between condition attributes and decision attribute can reflect the monotonic consistency. It indicates that the RMI can be used as a heuristic to generate an ordinal decision tree. Since order-preserving is the first-considered index for a monotonic decision tree, the ordinal mutual information used to select expanded features plays an essential role in building monotonic decision trees.

3 Algorithm of Monotonic Decision Tree for Interval Valued Attributes

3.1 Split Rule

In the process of generating an ordinal decision tree, the domain of an interval valued attribute is normally divided into two sections by a point value. A threshold value T can be selected for an interval valued attribute C (which takes values of a set of finite closed interval). The domain of C is divided into two branches which are represented as $C < T$ and $C \not< T$ according to the threshold value T . Let $C = [x, y]$, if $y < T$, we say $C < T$. Similarly, if $T < x$ or $T \in [x, y]$, we say $C \not< T$. The threshold value T is called splitting point.

3.2 Selection of Extended Attributes

Selection of Candidate Split Point. Suppose that there is a node with N training samples. The training samples are divided into K classes, labeled with C_1, C_2, \dots, C_K . For an interval valued attribute, the sample values are $e_i = [e_i^-, e_i^+](i = 1, 2, \dots, N)$. An ascending ordered sequence of $2N$ points can be obtained by sorting the endpoints of the N samples. The mean of any adjacent points in this sequence is considered as the candidate split point. There will be $2N - 1$ candidate split points for an interval valued attribute when the split points in a sequence are not repeated. The ordinal mutual information is then calculated according to each of the candidate split points.

Heuristic Information. Ranking mutual information is the degree of monotonicity between condition attributes and decision attribute. In ordinal classification, the monotonicity should be kept in classification learning.

Ranking mutual information can be used to reflect the monotonicity relevance between features and decisions. It is useful for ordinal feature selection and ordinal decision tree construction in ordinal classification, multicriteria decision making and ranking analysis [18]. So we select the ranking mutual information as the heuristic in this paper.

Approach to Extended Attribute Selection. The training samples are divided into two sets S_1 and S_2 according to a candidate split point T of one condition attribute B , where the interval values of samples in S_1 meet the condition $C < T$, in S_2 meet the condition $C \not< T$. Then the values of samples in S_1 are marked as 1 and values in S_2 are marked as 2. A value of RMI can be calculated based on each candidate splitting point of B and D , and the maximum value of RMI is selected as the rank mutual information between B and D . In this way we obtain a maximum value of RMI for each condition attribute. Computing the biggest one by making a comparison among these maximum values, we can select the condition attribute which is corresponding to the biggest value of RMI as the extended attribute. The candidate splitting point corresponding to the selected expanded attributes is then selected as the best split point.

Our Proposed Algorithm. The proposed algorithm is described as follows.
 Input: the left endpoint data sets of every training sample, the right endpoint data sets of every training sample, the decision attribute value of every sample.
 Output: monotonic decision tree for interval valued data.

(i) If the number of sample is 1 or all the samples come from the same class, the branch stops growing;

(ii) Otherwise

For each attribute a_i ($[e_i^-, e_i^+]$);

For each splitting point c_j ;

Divide samples into two subsets according to c_j ;

For each sample x_k ;

If $e_{ik}^+ < c_j$, then $a_{ki} = 1$;

Else $a_{ki} = 2$;

Endif;

Compute $RMI_{a_i, c_j} = RMI(a_i, D)$;

Endfor;

$c_{ij}^* = \arg \max_j \{ RMI_{a_i, c_j} \}$

Endfor;

Endfor;

(iii) Select the best feature and the corresponding splitting point with the following formula;

$$c^* = \arg \max_i \left\{ \max_j \{RMI_{a_i, c_j}\} \right\} \tag{8}$$

- (iv) Split samples with a and c^* and build a new node;
- (v) Recursively produces new splits according to the above procedure until stopping criterion is satisfied;
- (vi) End.

The decision tree stops growing when the samples in the node belong to the same class or there is only one sample in the node. Then the node is called leaf node, and it is labeled as the class of samples or sample in the current node.

4 An Example

In this section, we give an example to illustrate the inductive process using the small training data set shown in Table 3.

For each attribute $A_i (i = 1, 2, 3)$, the total number of splitting points is 48, if there are same splitting points, we remain one only. For attribute A_1 , the splitting points of A_1 and the corresponding RMI are shown in Table1.

Table 1. Splitting points and the corresponding RMI of A_1

splitting points	RMI
0.5	0
1.5	0.0575
2.5	0.0883
3.5	0.1545
4.5	0.1545
5.5	0.1902
6.5	0.3564
7.5	0.4054
8.5	0.4583
9.5	0.4690
10.5	0.2814
11.5	0.2345
12.5	0.1407
13.5	0.0469

From the table we can see the maximum RMI is 0.4690, its corresponding splitting point is 9.5000. Similarly, the maximum RMI of A_2 and A_3 are 0.3651 and 0.3564, their corresponding splitting points are 3.2500 and 1.1000 respectively. The value 0.4690 is the maximum RMI, so we select A_1 as the extended attribute, 9.5000 is the splitting point.

The sample set is partitioned by A_1 to two subsets: $\{x_2, x_4, x_6, x_8, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{20}, x_{22}, x_{23}, x_{24}\}$ and $\{x_1, x_3, x_5, x_7, x_9, x_{16}, x_{17}, x_{18}, x_{19}, x_{21}\}$. Repeated this process, a decision tree is generated as Figure 1.

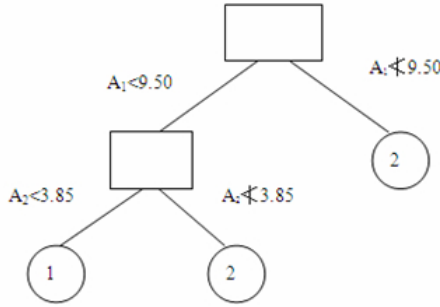


Fig. 1. The decision tree generated with the our method

Table 2. Test set with interval valued attributes

test samples	A_1	A_2	A_3	D
x_1	[6.00, 11.00]	[3.00, 5.00]	[0.50, 1.60]	2
x_2	[1.00, 4.00]	[1.00, 3.00]	[0.20, 0.80]	1
x_3	[[2.00, 8.00]]	[1.00, 3.50]	[0.10, 1.00]	1
x_4	[7.00, 12.00]	[3.00, 6.00]	[0.80, 1.50]	2

Here, we use 4 test samples shown in Table 2 to verify the monotonic character of the generated decision tree.

According to the above decision tree, x_1 is assigned into the right branch in the first level, and we can see the outputted decision attribute meets the expected result. Similarly, we can test x_2, x_3, x_4 . So we can verify the generated decision tree is monotonic.

In addition, according to the definition of monotonically consistent presented in [19], the above decision tree is monotonically consistent.

5 Conclusions

This paper attempts to address the problem of inductive learning a monotonic decision tree from the data sets with interval values, and proposes an algorithm to generate such kind of decision tree with the monotonically consistent property. The rank mutual information is employed to select the extended attributes in the proposed algorithm, and it can be guaranteed that the decision tree generated with the proposed algorithm is monotonic. An example is given to illustrate the process of generating and testing the decision tree for our proposed algorithm, which guarantees that the decision tree is ordinal.

Acknowledgments. This research is supported by the national natural science foundation of China (61170040, 71371063), by the key scientific research foundation of education department of Hebei Province (ZD20131028), by the scientific research foundation of education department of Hebei Province (Z2012101), and by the natural science foundation of Hebei Province (F2013201110, F2013201220).

References

1. Quinlan, J.R.: Induction of decision trees. *Machine Learning* **1**(1), 81–106 (1986)
2. Han, J., Kamber, M., Pei, J.,: *Data Mining: Concepts and Techniques*, Third Edition. Elsevier Inc. (2012)
3. Christopher, M.B.: *Pattern Recognition and Machine Learning*. Springer (2007)
4. Mitchell, T.M.: *Machine Learning*. McGraw-Hill Companies, Inc. (1997)
5. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. San Matco, Morgan Kaufmann (1993)
6. Cheng, J., Fayyad, U.M., Irani, K.B., et al.: Improved decision trees: a generalized version of ID3. In: Dietterich, T. (ed.) *Proceedings of the 5th International Conference on Machine Learning*, pp. 100–108. San Matyeo, Morgan Kaufmann Publishers (1988)
7. Quinlan, J.R.: Probabilistic decision trees. In: Kodratoff, Y., Michalski, R. (eds.) *Maching Learning: An Artificial Intelligence Approach*, pp. 140–152 Vol. 3. San Matyeo, Morgan Kaufmann Publishers. 3 (1990)
8. Hong, J.R., Ding, M.F., Li, X.Y., et al.: A new decision tree inductive learning algorithm. *Journal of computer* **18**(6), 470–474 (1995)
9. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* **8**, 87–102 (1992)
10. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. *Fuzzy Sets and Systems* **69**, 125–139 (1995)
11. Potharst, R., Bioch, J.C.: Decision trees for ordinal classification. *Intelligent Data Analysis* **4**(2), 97–111 (2000)
12. Frank, E., Hall, M.: A Simple Approach to Ordinal Classification. In: Flach, P.A., De Raedt, L. (eds.) *ECML 2001. LNCS (LNAI)*, vol. 2167, p. 145. Springer, Heidelberg (2001)
13. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. *European journal of operational research* **138**(2), 247–259 (2002)
14. Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht (1991)
15. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research* **117**(1), 63–83 (1999)
16. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation by dominance relations. *International journal of intelligent systems* **17**(2), 153–171 (2002)
17. Xia, F., Zhang, W., Li, F., et al.: Ranking with decision tree. *Knowledge and information systems* **17**(3), 381–395 (2008)
18. Hu, Q.H., Guo, M.Z., Yu, D.R., et al.: Information entropy for ordinal classification. *Science China Information Sciences* **53**(6), 1188–1200 (2010)
19. Hu, Q., Che, X., Zhang, L., et al.: Rank Entropy-Based Decision Trees for Monotonic Classification. *IEEE Transactions on Knowledge and Data Engineering* **24**(11), 2052–2064 (2012)

20. Wang, X.Z., Hong, J.R.: Interval valued attributes decision tree learning algorithm. *Journal of software* **9**(8), 637–640 (1998)
21. Liu, J., Liu, S.F.: Sort research for multiple attribute object with interval valued attributes. *Chinese management science* **18**(3), 90–94 (2010)
22. Liang, J.Y., Qian, Y.H.: Information granules and entropy theory in information systems. *Science in China Series F: Information Sciences* **51**(10), 1427–1444 (2008)
23. Hu, D., Li, H.X., Yu, X.C.: The information content of rules and rule sets and its application. *Science in China Series F: Information Sciences* **51**(12), 1958–1979 (2008)
24. Mingers, J.: An empirical comparison of selection measures for decision-tree induction. *Machine learning* **3**(4), 319–342 (1989)
25. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1226–1238 (2005)
26. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. *Machine learning* **8**(1), 87–102 (1992)
27. Viola, P., Wells III, W.M.: Alignment by maximization of mutual information. *International journal of computer vision* **24**(2), 137–154 (1997)
28. Spearman, C.: Footrule for measuring correlation. *British Journal of Psychology*, 1904–1920, **2**(1), 89–108 (1906)

Parallel Ordinal Decision Tree Algorithm and Its Implementation in Framework of MapReduce

Shanshan Wang, Junhai Zhai^(✉), Hong Zhu, and Xizhao Wang

College of Mathematics and Computer Science, Hebei University,
Baoding 071002, China
mczjh@126.com

Abstract. Ordinal decision tree (ODT) can effectively deal with monotonic classification problems. However, it is difficult for the existing ordinal decision tree algorithms to learning ODT from large data sets. In order to deal with the problem of generating an ODT from large datasets, this paper presents a parallel processing mechanism in the framework of MapReduce. Similar to the general ordinal decision tree algorithms, the rank mutual information (RMI) is still used to select the extended attributes. Differing from the calculation of RMI in the previous algorithms, this paper applies a strategy of attribute parallelization to calculate the RMI. Experiments on large ordered data sets (which are generated artificially) confirm that our proposed algorithm is feasible. Experimental results show that our algorithm is effective and efficient from three aspects: speed-up, scale-up and size-up.

Keywords: Monotonic classification · Ordinal decision tree · Rank mutual information · MapReduce

1 Introduction

Ordinal decision tree algorithms [1, 2] are the extensions of classical decision tree algorithms to deal with ordinal classification problems in which the condition attributes and the decision attribute are all ordered while the classification rules need to be order-preserving [1]. With the rapid development of computer networking including the wireless sensor and data storage technologies, large scale data sets can be easily collected in many fields. The size of large dataset usually becomes too big to process for a single computer in an acceptable time interval. Due to the limit of computational resource, it has been a big challenge for the existing decision tree algorithms to extract classification rules from very large data sets. It is even impracticable for decision tree algorithms with high computational complexity to learn the rules from the large data sets. In response to this challenge, as early as 1997, Kufirin [3] discussed the parallelization problem of decision tree, and proposed a parallelization framework for induction of decision trees. But as it is well known, in this age, the concept of large data sets is different from 1990s. In 2007, Olcay and Onur [4] proposed parallel implementations of two univariate

decision tree algorithms (C4.5 and linear discriminant tree). The proposed algorithms are parallelized by distributing the features, the data or the nodes among the slave processors. In 2009, Wu et al. [5] designed and implemented a bagging ensemble method name MReC4.5 using the Hadoop parallel and distributed computing model. Based on the programming framework-MapReduce, He et al. [6] proposed a parallel ID3 classification algorithm, and applied the proposed classification approach to the water quality analysis. Yin et al. [7] presented an open implementation of a scalable regression tree algorithm on Hadoop. Zhu et al. [8] studied the properties of Gini impurity, a measure for determining split points, and designed an algorithm for calculating split points with MapReduce. For the imbalanced big data learning problems, based on the random forest classifier, the oversampling, under-sampling and cost-sensitive learning technique were adapted to big data using MapReduce by Sara et al. [9].

The algorithms mentioned above are tailored for general classification problems which do not consider the order relations between conditional attributes and the decision attribute. Ordinal classification problems (also named monotonic classification problems) are another kind of classification problems. Ordinal classification problems can be used to describe a number of particular applications, such as, manuscript evaluation, scholarship evaluation, university rating, and risk evaluation, etc. Several ordinal decision tree algorithms have been proposed by different authors. For example, Potharst and Bioch [10] designed an order-preserving tree-generation algorithm for repairing non-monotonic decision trees; and Xia et al. [11] extended the Gini impurity used in CART to ordinal classification problems and then presented a splitting heuristic. Combing the advantage of information entropy with dominance rough sets, Hu et al. [12] gave the definitions of rank entropy and rank mutual information, which can be regarded as a new measure for ordinal information. This new measure can not only measure the monotonous consistency in monotonic classification, but also demonstrate the heuristics robustness noisy samples. Furthermore, based on mutual information, an algorithm for generating monotonic decision tree was proposed in [13].

From references, we do not find the induction of ordinal decision trees and their corresponding classification algorithms for very large datasets. This paper makes a first attempt to design a parallel algorithm for ordinal classification of big data. Under the framework of MapReduce, a parallel ordered decision tree algorithm named PODT is proposed and implemented in this paper. Similar to the general ordinal decision tree algorithms [13], PODT still uses the rank mutual information (RMI) as a heuristic to select the extended attributes. Differing from the calculation of RMI in the previous algorithms, the PODT applies a strategy of attribute parallelization to calculate the RMI.

The rest of the paper is organized as follows. Section 2 provides some necessary preliminaries including the MapReduce framework and a number of basic concepts of ordinal classification based on ordinal decision trees. Section 3 presents our parallel ordinal decision tree algorithm and its implementation under the framework of MapReduce. Section 4 lists our experimental results and their corresponding analysis. Section 5 concludes this paper.

2 Preliminaries

2.1 MapReduce

MapReduce [14] is a distributed computing framework introduced by Google for processing large data sets with parallel technique. Basically MapReduce consists of two phases: Map and Reduce, which are shown in Figure 1. The popular implementation of MapReduce is the Hadoop [15] initiated by Yahoo.

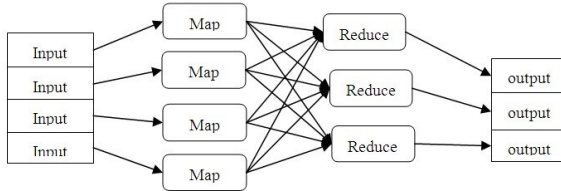


Fig. 1. Data manipulation processes with MapReduce

As shown in Figure 1, Map process partitions the input data set into M blocks, and assigns tasks to the Map respectively. How to partition the input data, which generally depends on the specific problem, will have a direct impact on the final result. Usually each Map reads data one by one and processes every data according to Mapper. One can run multiple Map tasks simultaneously on one node. Multiple Map tasks can run independently to each other. The input and output of MapReduce programming model are formed as key-value pairs. Mapper receives key-value (key pairs) as an input. Then we perform the Mapper to generate key-value (key pairs) as intermediate results. The intermediate results of nodes are sorted according to key values and the value of the same key are integrated in the Combiner process period. Then the results are sent to Reduce after treatment. The intermediate results are cut into several groups and pass to reduce operation according to the number of Reduce. Reduce function receives the intermediate results which are in the form of key-value pairs, processes the obtained value and gives the final output. Finally, the results will be output in the form of key-value pairs.

2.2 Ordinal Classification and Rank Mutual Information

Let U be a set of samples described with a set of attributes A and a decision variable D . Given $\forall a \in A$ and $\forall x, y \in U$, if we have $a(x) \geq a(y)$, we then say that x is better than y with respect to attribute a . For $B \subseteq A$ if $\forall a \in B$, the inequality $a(x) \geq a(y)$ holds well, then we denote this relation by $x \geq B_y$. In addition, suppose that there exists a partially ordered structure between the decision labels $D = w_1, w_2, \dots, w_n$, denoted by $w_1 \prec w_2 \prec \dots \prec w_n$. A classification function f is said to be monotone with respect to B if for $x, y \in U : x \geq B_y \rightarrow f(x) \geq f(y)$.

Given U , for $\forall x \in U, a \in B \subseteq A$, we can associate x with the following sets:[13]

$$[x]_a^{\geq} = \{y \in U : y \geq_a x\} \tag{1}$$

$$[x]_B^{\geq} = \{y \in U : y \geq_B x\} \tag{2}$$

$$[x]_a^{\leq} = \{y \in U : y \leq_a x\} \tag{3}$$

$$[x]_B^{\leq} = \{y \in U : y \leq_B x\} \tag{4}$$

Given $U, B, C \subseteq A$, the upwards ranking mutual information of B and C can be defined as: [13]

$$RMI^{\geq}(B, C) = -\frac{1}{|U|} \sum_{i=1}^n \log \frac{|[x_i]_B^{\geq}| \times |[x_i]_C^{\geq}|}{|U| \times |[x_i]_B^{\geq} \cap [x_i]_C^{\geq}|} \tag{5}$$

Based on the ranking mutual information, a decision tree algorithm named RMIDT was proposed in [13]. This is a typical work for generating ordinal decision tree by using the ranking mutual information. One disadvantage is that the computational complexity of the algorithm is high. It is not suitable for handling large-scale data sets. We list the RMIDT algorithm as follows.

Algorithm 1. RMIDT

input: Training Set T ; Attributes S ; Stop Rules c

output: Ordinal decision tree TR

- 1: If T is NULL then Return failure;
 - 2: If S is NULL then Return the maximum decision attribute as the label;
 - 3: If $S=1$ then Return S as the label for the Tree;
 - 4: Set $TR=$;
 - 5: For a in S do;
 - 6: Set $RMI=0$ Compute RMI;
 - 7: For each v in values(a, T) do;
 - 8: To discrete the attribute value of T according to v ;
 - 9: For each x in T do;
 - 10: If $a(x)<v$ then $a(x)=1$;
 - 11: Else $a(x)=2$;
 - 12: $RMI(v, a)=RMI(S, T)$;
-

3 Parallel Ordinal Decision Tree Based on MapReduce

In this section, we will present our parallel scheme for ordinal decision tree in big data. Mainly the parallel algorithm and its implementation includes of two phases: data preparation, RMI calculation and optimal attribute selection.

3.1 Data Preparation

Data preparation mainly arranges the input dataset into pairs like $\langle(\text{condition attribute index, decision attribute value), condition attribute value}\rangle$ where the

first item (condition attribute index, decision attribute value) is corresponding to the key in MapReduce while the second item condition attribute is corresponding to the key value. And then the input dataset is stored into the HDFS (Hadoop Distributed File System) as a sequence file of (key, value) pair. Each pair represents a sample in the dataset. The input key of Mapper is the offset in bytes of each sample to the start point of the data file and the input value of Mapper is the content of the sample, including condition attributes and decision attributes. Samples are randomly assigned to each Mapper. Because the operation requires all values of conditional attributes and decision attribute, the intermediate output key is designed in the form of (condition attribute index, decision value) pairs, and the intermediate output value is the conditional attribute values in subsequent operations.

Algorithm 2 shows the map operation of MapReduce on the data preparation phase. Mapper mainly reads every attribute value of each sample, and then outputs the result in accordance with the requirements. Algorithm 3 is a combiner operation and its main purpose is conducting statute in the Mapper terminal though it reduces the number of network transmissions. We consider the output (key, value) pair is designed as follows: the output key is the conditional attribute index, and the output value is the set of all conditional attribute values and the decision values in the Mapper terminal. [We consider the output (key, value) pair is designed as follows: the output key is the conditional attribute index, and the output value is the set of all conditional attribute values and the decision values in the Mapper terminal.]This design reduces the amount of communications of Mapper and Reducer terminals. The pseudo codes of Algorithms 2 and 3 are listed as follows:

Algorithm 2. Prepare Mapper (key, value)

input: (key, value) pair, key is the offset in bytes, value is the content of a sample.
output: (key, value) pair, where key is the content of condition attribute index and the decision attribute values, value is the value of the attribute.

- 1: Parse the value to an array, named sample;
- 2: If S is NULL then Return the maximum decision attribute as the label;
- 3: Read the value of input, which contain each attribute of a sample, and count its number named Num;
- 4: Initiate string outkey as a null string, used to output the key value;
- 5: Initiate string outValue as a null string, used to output the value value;
- 6: For i=1 to Num-1 do;
- 7: Initiate string tempString as a null string;
- 8: tempString = sample [i]; //read the ith attribute value
- 9: outValue.append(tempString);
- 10: output(outKey,outValue);
- 11: Endfor.

In Algorithm 2, Steps 1-4 are some preparation for the calculating the key value pairs, including parsing the input values; initiating variable keys; and getting the values for intermediate outputs. Steps 5-9 identify (condition attribute

index, decision attribute value) pair as the output key value and identify condition attribute value as the output value. Steps 10-11 are to output the intermediate (key, value) pair, denoted by (outKey, outValue).

Algorithm 3. Prepare Combiner (key, value)

input: (key, value) pair, where key is the condition attribute index and decision attribute value, value is a string containing the values of the condition attribute.
output: (key, value) pair, where key is the index of an conditional attribute, value is a string containing decision attribute value and conditional attribute value

- 1: Parse the key as a string, read condition attribute index and decision attribute value separately, named keystack and decstring;
- 2: Parse the value to an array, named sample;
- 3: Initiate string outkey as a null string, used to output the key value;
- 4: Initiate string outValue as a null string, used to output the value value;
- 5: outValue.append(decstring);
- 6: For i=0 to sample.size-1 do;
- 7: outValue.append(sample[i]);
- 8: Endfor;
- 9: outKey.append(keystack);
- 10: Output(outkey,outValue).

In Algorithm 3, steps 1-4 are the initialization for computing (key, value) pair, steps 5-8 package the decision values and conditional attribute values; and steps 9 and 10 are to output the intermediate (key, value) pair, (outKey, outValue).

Algorithm 2 and algorithm 3 read samples from HDFS and package the data according to a certain output format, and then send the data to reduce terminal. Algorithm 4 receives the data which are managed in algorithm 2 and algorithm 3, calculates RMI and selects the optimal attribute.

3.2 RMI Calculation and Optimal Attribute Selection

The following Algorithm 4 is the essential part of our proposed approach, which mainly makes a parallelization of computation among attributes. Reduce function calculates the maximum RMI of individual attributes, designs condition attribute index as the input key value, and lets the intermediate results received from the Mapper terminal as the input value. The reduce result is the maximum RMI for the considered attribute. Algorithm 4 gives the pseudo code of the reduce function in MapReduce for calculating RMI:

In Algorithm 4, step 1 initiates variables used to storage each value of ranking mutual information, to storage the maximum value of ranking mutual information, and to storage the optimal cut point. Steps 2-6 read the input value and steps 7-9 calculate ranking mutual information of each example. Steps 10-13 make a comparison among optimal cut points according to the RMI and then take the maximum. Steps 14-16 are to output the value (key, value) pair.

Algorithm 4. AttributeReducer(key, value)

input: (key, value) pair, where key is the index of a condition attribute, value is a string of the intermediate results.

output: (key, value) pair, where key is a random string, I defined its value as RMI, value is a string containing condition attribute index, the best optimal split point and the maximum value of the RMI.

- 1: Initiate a variable RmiResult,TmpResult,BetterVaue = 0;
- 2: Normalize the dependent variables of each input data named;
- 3: For each value in input value
- 4: Parse the single value to an array, named line;
- 5: Read line, which contain condition attribute values named AttrValue and decision attribute value named DecValue, assigned to array TempVal and CopyVal;
- 6: End for;
- 7: For each attribute value named val do;
- 8: For i=0 to TempVal.size()-1 do
- 9: Calculate RMI based on Hu's theory, described in Algorithm 1,and RmiResult keeps the RMI result;
- 10: if RmiResult >= TmpRmiResult;
- 11: TmpRmiResult = RmiResult;
- 12: BetterValue = val.AttrValue;
- 13: End for;
- 14: outKey.append(RMI);
- 15: outValue.append(key+BetterValue+TmpRmiResult);
- 16: output(outkey,outValue).

4 Experiments

The proposed algorithm is implemented on a parallel system. The hardware environment is a computer cluster with six nodes and each node is a PC with 2.5GHz CPU and 4G memories. The software environment is in the framework of Hadoop MapReduce. The performance of the proposed algorithm is evaluated in three aspects: Speedup, Scaleup and Sizeup [6, 16], which are defined as follows: $Speedup(m) = \frac{t_1^{(1)}}{t_m^{(1)}}$, $Scaleup(m) = \frac{t_1^{(2)}}{t_m^{(2)}}$, $Sizeup(m) = \frac{t_1^{(3)}}{t_m^{(3)}}$ where $t_1^{(1)}$ is the computation time on one node with all attributes while $t_m^{(1)}$ is the computing time on m nodes with all attributes. $t_1^{(2)}$ is the computation time on one node with one attribute while $t_m^{(2)}$ is the computing time on m nodes with m attributes. $t_1^{(3)}$ is the computation time on one clouding platform (one cluster) with one attribute while $t_m^{(3)}$ is the computation time on one clouding platform (one cluster) with m attributes.

In order to show the effectiveness of our proposed parallel algorithm, we conduct some experiments with artificial data sets which proposed in Hu paper [13]. First, we generated a big monotone data sets based on the function $f(x_1, x_2) = x_1 + \frac{1}{2}(x_2^2 - x_1^2)$, where x_1 and x_2 are two random variables independently.

In order to generate ordered class labels, the resulting numeric values were discretized into k intervals $[0, 1/k], (1/k, 2/k], \dots, (k - 1/k, 1]$. Thus each interval contains approximately the same number of samples. The samples belonging to

one of the intervals share the same rank label. Then we form a k -class monotonic classification task. In this experiment, we try $k=4, 6, 8$ and 12 , respectively.

Because we need multiple attributes, we used the function to generate attributes repeatedly and several classes.

Speedup refers to how faster a parallel algorithm with m processors than a serial algorithm. To measure the Speedup, we keep the number of attributes constant and increase the number of cores in the system. We have performed the speedup evaluation on datasets with different number of attributes and keep the size of dataset 2 million [6]. The number of computers is 2 to 6 respectively. The number of attributes varies from 4, 6 to 10. Figure 2 shows the experimental results. As the number of attributes increases, the Speedup performs better.

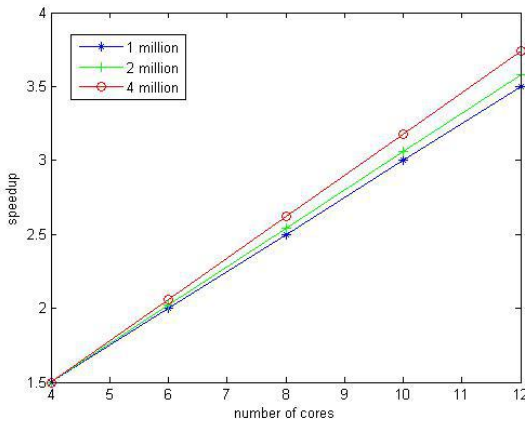


Fig. 2. The experimental results on the Speedup

Scaleup describes the extended performance of the algorithm. The performance of the algorithm is improved with the increasing both the number of attributes and the number of computing cores (while the number of attributes are increased proportionately with the computing cores). The experimental results show that the proposed algorithm is suitable for constructing large multi-attribute ordinal decision tree. We have performed scalability experiments where we increase the number of attributes in proportion to the number of cores. The data set size keeps in 2 million. Suppose N attributes are performed on N cores while N equals 4,8,12. Figure 3 shows the performance of our algorithm.

Sizeup describes time trends as the amount of attribute increasing. We keep the number of cores constant and grow the number of attributes. To measure the performance of size-up, we have fixed the number of cores to 4, 8 and 12 respectively. Figure 4 shows the experimental result. From Fig. 4 one can see that, with the increase of the number of attributes, the algorithm demonstrates a better performance for the Sizeup.

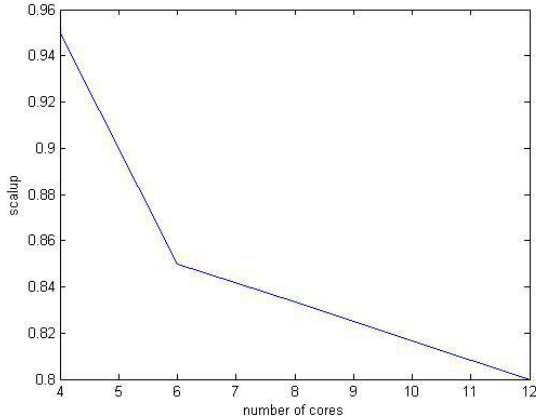


Fig. 3. The experimental results on the Scalup

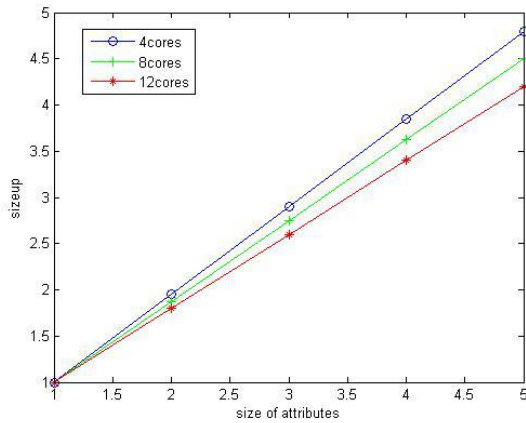


Fig. 4. The experimental results on the Sizeup

5 Conclusions

This paper attempts to address the problem of learning an ordinal decision tree from large data sets. In the framework of MapReduce, we design and implement a parallel ordinal decision tree algorithm based on ranking mutual information (shorted as PODT). A number of experiments on artificially generated large data sets with orders are conducted. Experimental results demonstrate a good performance of parallelization for our designed algorithms. Furthermore the paper shows an essential difference between the ordinal decision tree and general

decision tree (like C4.5) inductions from aspects of both the algorithm design and the implementation. Our future work regarding this topic will include:

- (1) the parallel selection of expanded attributes with respect to ordinal mutual information in the framework of MapReduce;
- (2) the improvement of parallelization of the ordinal decision tree induction for very large data sets;
- (3) the sample selection mechanism based on uncertainty reduction and the ordinal mutual information criterion.

Acknowledgments. This research is supported by the national natural science foundation of China (61170040, 71371063), by the key scientific research foundation of education department of Hebei Province (ZD20131028), by the scientific research foundation of education department of Hebei Province (Z2012101), and by the natural science foundation of Hebei Province (F2013201110, F2013201220).

References

1. Potharst, R., Bioch, J.C.: Decision trees for ordinal classification. *Intelligent Data Analysis* **4**(2), 97–111 (2000)
2. Hu, Q.H., Guo, M.Z., Yu, D.R., et al.: Information entropy for ordinal classification. *Science China Information Sciences* **53**(6), 1188–1200 (2010)
3. Kufirin, R.: Decision trees on parallel processors. *Machine Intelligence and Pattern Recognition* **20**, 279–306 (1999)
4. Olcay, T.Y., Onur, D.: Parallel univariate decision trees. *Pattern Recognition Letters* **28**, 825–832 (2007)
5. Wu, G., Li, H., Hu, X., et al.: MReC4.5: C4.5 ensemble classification with MapReduce. *The Fourth ChinaGrid Annual Conference*, 249–255 (2009)
6. He, Q., Dong, Z., Zhuang, F., Shang, T., Shi, Z.: Parallel Decision Tree with Application to Water Quality Data Analysis. In: Wang, J., Yen, G.G., Polycarpou, M.M. (eds.) *ISNN 2012, Part II. LNCS*, vol. 7368, pp. 628–637. Springer, Heidelberg (2012)
7. Yin, W., Simmhan, Y., Prasanna, V.K.: Scalable regression tree learning on Hadoop using OpenPlanet. *Proceedings of third international workshop on MapReduce and its Applications*. Date, 57–64 (2012)
8. Zhu, M., Shen, D., Yu, G., et al.: Computing the Split Points for Learning Decision Tree in MapReduce. *Database Systems for Advanced Applications, Lecture Notes in Computer Science* **7826**, 339–353 (2013)
9. Sara, R., Victoria, L., Jos, M., et al.: On the use of MapReduce for imbalanced big data using Random Forest. *Information Sciences 2014.03.043* (2014)
10. Potharst, R., Bioch, J.C.: Decision trees for ordinal classification. *Intelligent Data Analysis* **4**(2), 97–111 (2000)
11. Xia, F., Zhang, W., Li, F., et al.: Ranking with decision tree. *Knowledge and information systems* **17**(3), 381–395 (2008)
12. Hu, Q.H., Guo, M.Z., Yu, D.R., et al.: Information entropy for ordinal classification. *Science China Information Sciences* **53**(6), 1188–1200 (2010)

13. Hu, Q., Che, X., Zhang, L., et al.: Rank Entropy-Based Decision Trees for Monotonic Classification. *IEEE Transactions on Knowledge and Data Engineering* **24**(11), 2052–2064 (2012)
14. Jeffrey, D., Sanjay, G.: MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1), pp. 107–113 (January 2008)
15. <http://hadoop.apache.org/>
16. He, Q., Shang, T.: Parallel extreme learning machine for regression based on MapReduce. *Neurocomputing* **102**, 52–58 (2013)

Learning and Adaptation

An Improved Iterative Closest Point Algorithm for Rigid Point Registration

Junfen Chen^(✉) and Bahari Belaton

School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
chenjunfenusm@gmail.com, bahari@webmail.cs.usm.my

Abstract. Iterative Closest Point (ICP) is a popular rigid point set registration method that has been used to align two or more rigid shapes. In order to reduce the computation complexity and improve the flexibility of ICP algorithm, an efficient and robust subset-ICP rigid registration method is proposed in this paper. It searches for the corresponding pairs on subsets of the entire data, which can provide structural information to benefit the registration. Experimental results on 2D and 3D point sets demonstrate the efficiency and robustness of the proposed method.

Keywords: Iterative Closest Point (ICP) · Point registration · Rigid transformation · Subset · Alignment

1 Introduction

Point set registration is an important research topic in computer vision and image processing. It has been widely applied to pattern recognition, shape reconstruction, motion tracking and stereo matching, etc. The task of point set registration is to recover an optimal transformation according to the current locations of two point sets and map one point set onto another to make them overlap as much as possible. Simultaneously, the shapes described by the points are aligned very well as shown in Figure 1.

Iterative Closest Point (ICP) algorithm [1, 2] is one of the most popular rigid point set registration methods. Expectation Maximization (EM) scheme [3] is often used as the alternating update procedure to search for the solution, whose E step and M step can be viewed as updating correspondences and recovering transformation of ICP respectively. An EM-ICP [4] was proposed to handle Gaussian noise in rigid registration of large points set. It was a coarse-to-fine approach based on an annealing scheme to balance the robustness and the accuracy of ICP. Liu [5] combined the soft-assign and EM-ICP algorithms for the automatic registration of overlapping 3D point clouds. When the transformation (M step) is required to be determined with fix correspondences, multiple layer feed-forward neural network [6] is an alternative rigid point set registration method. Combined with Principal Component Analysis (PCA) feature extraction, neural network can be used to align two rigid 2D gray images [7].

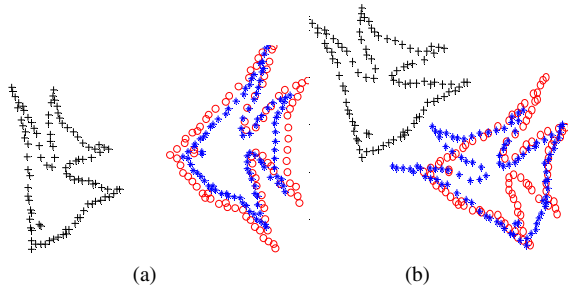


Fig. 1. Two registration results. (a) is successful because the distance measurement is small and the shapes are aligned. (b) is fail in spite of the smaller distance measurement but poor shape alignment.

Point set registration can be considered as a probability density estimation problem. Many probabilistic methods, such as Gaussian Mixture Model (GMM) [8, 9] and Robust Point Matching (RPM) [10], are developed to acquire better registration result in the presence of noise and outliers at the cost of computation complexity.

In this paper, an efficient and robust rigid point set registration method, named subset-ICP, is presented, which is an improved ICP method. In this method, a whole target set is divided into several subsets, and the same process is done to the source set. For each pair of the target subset and source subset, standard ICP is conducted to find the optimal transformation, which is used to map the entire source set to the target set.

The proposed method can deal with larger rotation very well. Partial data instead of entire data set can implicitly provide structural information. Under the viewpoint of optimization, with basic matrix theory, the induction procedure of the rigid transformation parameters (scaling, rotation and translation) is provided in this paper. It is easier to understand the proof procedure than that of the quaternion method based on PCA.

The rest of this paper is organized as follows. Fundamental knowledge is briefly described in section 2. Section 3 presents the proof of the unknown variables of rigid transformation, and introduces our subset-ICP method. Experiments are shown in section 4. Finally, conclusions are given in section 5.

2 Preliminary

In this section, the fundamental steps of standard ICP is introduced firstly and then some basic concepts about matrix theory are recalled to easily understand the induction procedure of section 3.

2.1 Standard ICP

Given two point sets $X = \{x_1, x_2, \dots, x_M\}$ and $Y = \{y_1, y_2, \dots, y_N\}$, where $x_i, y_j \in \mathbb{R}^n$ (each element is defined in n -dimensional Euclidean space), M, N are the numbers of points in X and Y respectively. The main steps of the standard ICP are the

correspondences and the transformation, which are updated till the terminate conditions are satisfied.

- For each point y_j ($j = 1, 2, \dots, N$) of set Y , search for its closest point x_i from set X to form the correspondences set $N_Y(X) = \{x_i \mid d(y_j, x_i) = \operatorname{argmin}_{x \in X} d(y_j, x)\}$;
- For sets $N_Y(X)$ and Y , compute the rotation matrix R^k and translation vector t^k using statistics technique PCA.
- Apply transformation (R^k, t^k) to update set Y and accumulate rotation matrix R and translation vector t .

$$Y = R^k \cdot Y + t^k \tag{1}$$

$$R = R^k \cdot R \tag{2}$$

$$t = R^k \cdot t + t^k \tag{3}$$

2.2 Basic Concepts About Matrix Theory

Definition 2.1 (vector inner product and vector norm): Given two n -dimensional vectors $a = (a_1, a_2, \dots, a_n)$ and $b = (b_1, b_2, \dots, b_n)$, then the squared Euclidean distance between a and b is rewritten using the inner product and norm of vector as

$$\|a - b\|^2 = \|a\|^2 + \|b\|^2 - 2(a, b) \tag{4}$$

It is easy to know (here, a, b are row vectors)

- ① $\|a\|^2 = (a, a) = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2} = aa^T$
- ② $(a, b) = (b, a)$
- ③ $(ka, b) = (a, kb) = k(a, b) = k \cdot ab^T$

Definition 2.2 (orthogonal matrix): Square matrix A is called as an orthogonal matrix when $AA^T = A^T A = I$ is satisfied, where I is an identity matrix.

Definition 2.3 (matrix trace): Let $A = (a_{ij})_{n \times n}$ is a square matrix, the trace of A is defined as

$$\operatorname{tr}(A) = \sum_{i=1}^n a_{ii} \tag{5}$$

The properties of trace are included:

- ① $\operatorname{tr}(A) = \operatorname{tr}(A^T)$
- ② $\operatorname{tr}(AB) = \operatorname{tr}(BA)$

3 Subset-ICP Registration Method

In this section, a proof of the unknown variables of rigid transformation is provided and then our subset-ICP method is introduced.

3.1 Parameters of the Rigid Transformation

The point set registration is to find an optimal mapping F to make $\sum \|x - F(y)\|^2$ as smaller as possible. When the rigid transformation is considered, the point set registration problem is transferred as a minimum optimization with constraints

$$\begin{aligned} \min Q(s, R, t) &= \sum ||x - (sRy + t)||^2 \\ \text{Subject to} \quad &R^T R = I \\ &\det(R) = 1 \end{aligned} \tag{6}$$

In order to obtain the closed form solution of (6), a lemma 1 [11] is described as follows:

Lemma 1: Let $R_{D \times D}$ be an unknown rotation matrix and $A_{D \times D}$ be a known real square matrix. Let USV^T be a Singular Value Decomposition of A, where $UU^T = VV^T = I$ and $S = d(s_i)$, with $s_1 \geq s_2 \geq \dots \geq s_D \geq 0$. Then, the optimal rotation matrix R that maximizes $tr(A^T R)$ is $R = UCV^T$, where $C = d(1, 1, \dots, 1, \det(UV^T))$.

For simplicity, we assume $X = \{x_1, x_2, \dots, x_M\}$ and $Y = \{y_1, y_2, \dots, y_N\}$, $M = N$ that denotes no outliers existing and the correspondences are established well. So the objective function is $Q(s, R, t) = \sum_{i=1}^N ||x_i - (sRy_i + t)||^2$.

Firstly, a partial derivative of Q with respect to t is computed and makes it equal to zero

$$\frac{\partial Q(s, R, t)}{\partial t} = -2 * \sum_{i=1}^N (x_i - sRy_i - t) = 0$$

We can obtain

$$t = \bar{x} - sR\bar{y} \tag{7}$$

where, the mean vector $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$, $\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$.

Based on formulae (7) and (4)

$$\begin{aligned} \sum_{i=1}^N ||x_i - (sRy_i + t)||^2 &= \sum_{i=1}^N ||(x_i - \bar{x}) - sR(y_i - \bar{y})||^2 = \sum_{i=1}^N ||\hat{x}_i||^2 + \\ s^2 \sum_{i=1}^N ||R\hat{y}_i||^2 - 2s \sum_{i=1}^N (\hat{x}_i, R\hat{y}_i) &= \sum_{i=1}^N (\hat{x}_i^T \hat{x}_i) + s^2 \sum_{i=1}^N (\hat{y}_i^T R^T R \hat{y}_i) - \\ 2s \sum_{i=1}^N (\hat{x}_i^T R \hat{y}_i) \end{aligned}$$

The objective function is written with matrix trace form as follows:

$$Q(s, R, t) = tr(\hat{X}^T \hat{X}) + s^2 tr(\hat{Y}^T \hat{Y}) - 2str(\hat{X}^T \hat{Y} R^T) \tag{8}$$

where, the following notations are used $\hat{x}_i = x_i - \bar{x}$, $\hat{y}_i = y_i - \bar{y}$, $\hat{X} =$

$$[\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N], \hat{X}^T = \begin{bmatrix} \hat{x}_1^T \\ \hat{x}_2^T \\ \vdots \\ \hat{x}_N^T \end{bmatrix}_{N \times n}, \hat{Y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N], \hat{Y}^T = \begin{bmatrix} \hat{y}_1^T \\ \hat{y}_2^T \\ \vdots \\ \hat{y}_N^T \end{bmatrix}_{N \times n}, n \text{ is the}$$

dimensional number of point.

The first two terms of (8) are independent of rotation matrix R, we can denote them as a constant c_2 . In addition, according to the property of trace, the last term of (8) is calculated as

$$tr(\hat{X}^T \hat{Y} R^T) = tr(((\hat{X}^T \hat{Y}) R^T)^T) = tr(R(\hat{X}^T \hat{Y})^T) = tr((\hat{X}^T \hat{Y})^T R) \tag{9}$$

Thus, the objective function can be converted as

$$Q(s, R, t) = -c_1 tr((\hat{X}^T \hat{Y})^T R) + c_2 \tag{10}$$

Based on Lemma1, let $A = \hat{X}^T \hat{Y}$, $USV^T = \text{svd}(A)$, the optimal rotation R is

$$R = UCV^T \tag{11}$$

where $C = d(1,1, \dots, 1, \det(UV^T))$.

In order to solve scaling factor s, taking a partial derivative of (8) with respect to s and make it equal to zero, then we have

$$s = \frac{tr(\hat{X}^T \hat{Y} R^T)}{tr(\hat{Y}^T \hat{Y})} \tag{12}$$

3.2 The Subset-ICP Algorithm

A simulated annealing combined with ICP scheme is used to improve the registration accuracy, at the same time, reduce the computation cost. The number of subsets is defined as a special temperature parameter T, where the increment is one for each iteration. Subset-ICP is implemented on two subsets to search for the closest points and to build the transformation mapping. The obtained mapping is employed to update the source data set Y. The pseudo-code of subset-ICP is described as follows in Figure 2:

```

Inputs: Point sets X and Y
Initialization: The scaling factor  $s_0 = 1$ , the rotation matrix  $R_0 = I$ 
               and the translation vector  $t_0 = 0$ 
For iterative registration
For simulated annealing ( $T = 1: \max\_T$ )

To update correspondences on subsets  $X_T, Y_T$ ;
To update transformation  $s, R$  and  $t$ ;
To apply the transformation to Y;

End simulated annealing
End iterative registration

Outputs: The transformation  $s, R$  and  $t$ ; The matched point set Y
    
```

Fig. 2. The pseudo-code of subset-ICP algorithm

4 Experiments

In this section, the experiments were conducted to verify the effectiveness of the proposed method in different kinds data including 2D fish clean points, 2D fish points with Gaussian noise, 3D bunny clean points, 3D bunny points with outliers points which can deteriorate the shape and 3D bunny data with missing data that means the incomplete data.

There are two point sets extracted from an object image, one is called the target set and its shape is represented by red circle, another is called source set and its shape is described by blue pentagrams. The task is to find an optimal spatial transformation (rotation matrix R and translation vector t in our experiments) according to the current

locations of the target set and the source set. If these two sets can be aligned very well (overlapping totally), then the registration (matching) is successful. The performance measures are Mean Squared Error (MSE) and visual results of registered images.

4.1 The Efficiency of Subset-ICP

2D fish data [12]: Four source point sets are synthetically formed by different linear transformations, blurring and deformation of the target point set. The registration results are shown in Figure 3. We can know that the proposed method is efficient for different spatial positions of rigid source point set, involving the larger rotation data. It is also robust to the noise data but not very effective to the deformable shape.

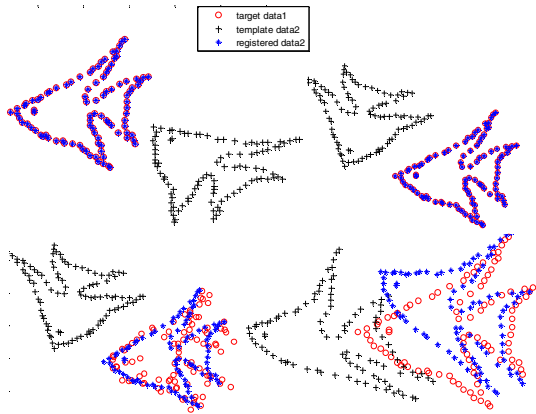


Fig. 3. The registration results of subset-ICP on 2D fish data. Red circle points form the target data1, black cross points are source data2 and the blue pentagram marks are the registered source data2

3D bunny data: we test our method on the stanford bunny data set [13]. 305 points are located manually to get the profile of 3D frontal bunny to form the target data1. Figure 4 demonstrates the qualitative registration results of subset-ICP on 3D bunny data, which validate the subset-ICP is an efficient rigid registration method.

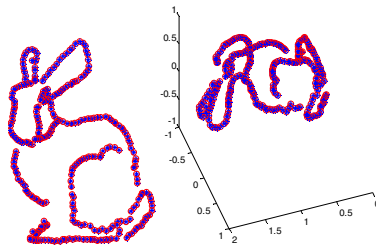


Fig. 4. The bunny data is registered by the subset-ICP. Left figure is a 2D result and the right figure is a 3D result

Missing data: The target bunny (red circle) and the source bunny (blue pentagram marks) are or/both with missing data in different parts respectively. The registration results are shown in Figure 5. We can see that the subset-ICP is robust to missing data.

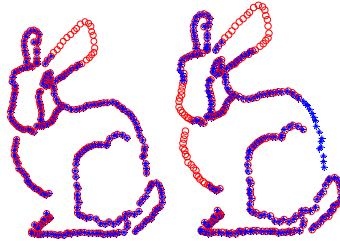


Fig. 5. The registration results of the subset-ICP when the missing data arising. The source data with missing ear data is in the left figure. The target data and the source data are both with missing data in different parts in the right figure

Outliers: The target data with 40% Gaussian outliers, the matching result is displayed in the left part of Figure 6. The source data with 20% Gaussian outliers, the matching result is shown in the right part of Figure 6. It is easy to see that the target with outliers do not affect the matching result but a slight difference exists when the source shape is disturbed by outliers. According to the experimental observation, for either the target shape or the source shape, an ill-matching happens when the structure information of shape is deteriorated by the outliers.

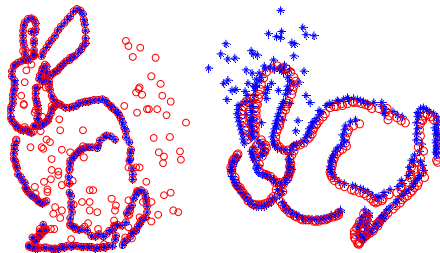


Fig. 6. The 2D matching results for the outliers. Target data with Gaussian outliers is shown in the left figure and the source data with Gaussian outliers is displayed in the right figure

4.2 The Comparison with Standard ICP

The comparisons between standard ICP algorithm and our method are displayed in Figure 7 and 8. The x-axis is the experiment times, that means ten pairs of the target sets and the source sets are generated to test the performances of subset-ICP and standard ICP algorithm. Y-axis of Figure7 is the MSE and y-axis of Figure8 is the speed (execution time (second)). Based on the experimental results, we can know that subset-ICP is faster and more efficient than standard ICP on 3D bunny data.

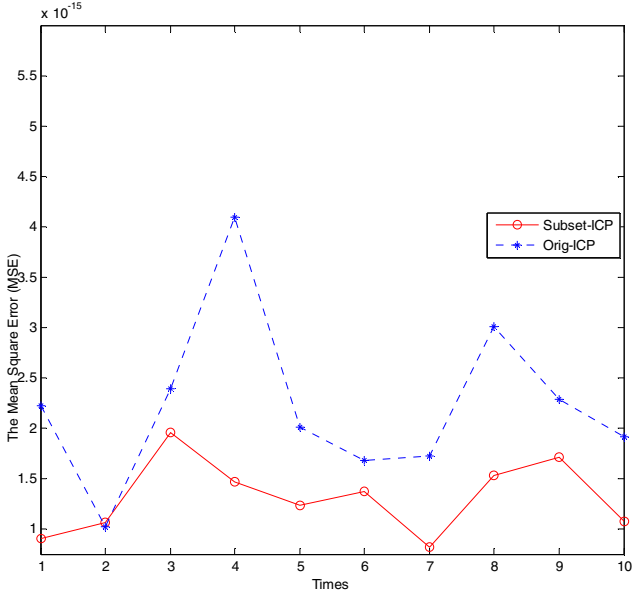


Fig. 7. The comparison of MSE on 3D bunny data

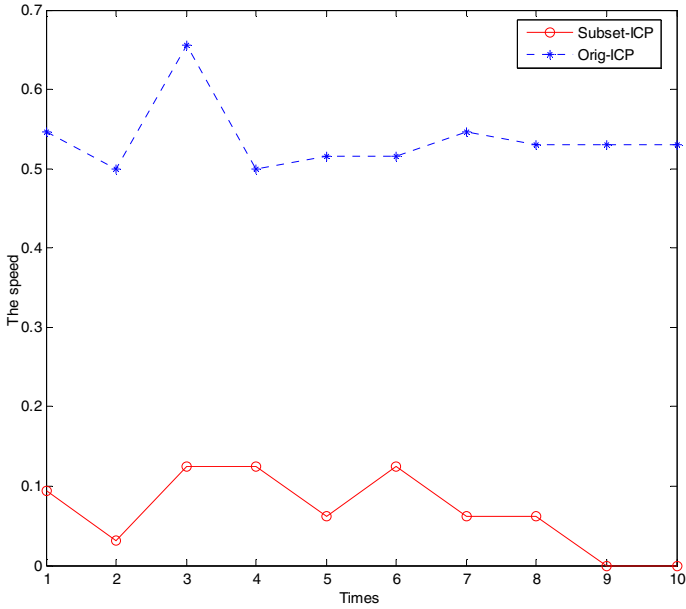


Fig. 8. The comparison of execution speed on 3D bunny data

5 Conclusions

In this paper, a subset-ICP rigid point set registration method is proposed, which is an improved version of standard ICP. Partial data instead of entire data can implicitly offer structural information that benefits the registration. Experiments are conducted on 2D and 3D synthetic data, and the matching results are analyzed and compared. All experimental results showed that the proposed method is efficient and robust. Under the viewpoint of optimization and matrix theory, the induction procedure of the rigid transformation parameters is also provided in this paper.

Acknowledgements. This work was supported by research project of Universiti Sains Malaysia (Grant: 1001/PKOMP/817055) and Natural Science foundation of Hebei Province (Grant: F2012201023).

References

1. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (1992)
2. Chen, Y., Medioni, G.: Object Modeling by Registration of Multiple Range Images. *Image and Vision Computing* **10**(3), 145–155 (1992)
3. McLachlan, G.J., Krishnan, T.: *The EM algorithm and extensions*. John Wiley & Sons, New York (1996)
4. Granger, S., Pennec, X.: Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part IV*. LNCS, vol. 2353, pp. 418–432. Springer, Heidelberg (2002)
5. Liu, Y.: Automatic registration of overlapping 3D point clouds using closest points. *Image and Vision Computing* **24**(7), 762–781 (2006)
6. Chen, J., Liao, I.Y., Belaton, B., Zaman, M.: A Neural Network-Based Registration Method for 3D Rigid Face Image. *World Wide Web* (2013) doi:10.1007/s11280-013-0213-9
7. Chen, J., Belaton, B., Pan, Z.: Multiple Components-Based 2D Face Images Alignment Using Neural Networks. *International Journal of Computational Science and Engineering* (2013) (under review)
8. Myronenko, A., Song, X.: Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(12), 2262–2275 (2010)
9. Rasouliyan, A., Rohling, R., Abolmaesumi, P.: Group-Wise Registration of Point Sets for Statistical Shape Models. *IEEE Transaction on Medical Imaging* **31**(11), 2025–2034 (2012)
10. Gold, S., Rangarajan, A., Lu, C.-P., Mjolsness, E.: New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence. *Pattern Recognition* **31**, 957–964 (1997)
11. Myronenko, A., Song, X.: On the Closed-Form Solution of the Rotation Matrix Arising in Computer Vision Problems. Presented at the Technical Report arXiv:0904.1613v1, Oregon Health and Science University (2009)
12. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding* **89**, 114–141 (2003)
13. The Stanford 3D Scanning Repository (2010). <http://graphics.stanford.edu/data/3Dscanrep/>

Approachs to Computing Maximal Consistent Block

Xiangrui Liu and Mingwen Shao^(✉)

Computer Engineering Institute, Qingdao Technological University,
Qingdao 266520, Shandong, People's Republic of China
lxr3014@126.com, mingwenshao@gmail.com

Abstract. Maximal consistent block is a technique for rule acquisition in incomplete information systems. It was first proposed by Yee Leung and Deyu Li in 2001. However, the maximal consistent blocks of an incomplete information system must be computed before they are put into use. In this paper, we introduced several approaches for computing maximal consistent block and their characteristics were further investigated. Each approach's time complexity is provided as well.

Keywords: Maximal consistent block · Rough sets · Incomplete information system

1 Introduction

Rough set theory, an innovation math theory which is capable of dealing with vague, uncertain and incompleted data, was first proposed by Z.Pawlak in 1982 [3]. In recent years, as it is widely used in machine learning and knowledge acquisition, data mining, decision support etc.(see [10,12,15,17,20]), people are starting to get interested in this area.

Rough set theory is based on classification of a set of objects in an information system. A classification is formed by a relation which can be represented by an indiscernibility relation. After classification, each set that has been divided by the relation is called knowledge. Rough set theory is the theory which shows us how to express some other sets by using the knowledge we have (divided by the relation). Apparently, this kind of description approximated as knowledge we have is limited.

Maximal consistent block is an approach in dealing with incomplete information system. Just as its name implies, an incomplete information system contains missing data (null values). That is to say a null value can be some values in the domain of the corresponding attribute. As an incomplete information system is more relevant to real life, a lot of work on it has been done. Some earlier techniques are to delete all objects that contain unknown values. It will change the structure of the original system, so an approach without changing the size of an information system was proposed in [4, 5]. Besides, different forms of incomplete information systems such as interval-valued, ordered information systems were also discussed in [8, 10, 12, 13, 14, 15, 21].

Yee Leung and Deyu Li proposed the innovative method, maximal consistent block technique [1]. It's a set whose objects are indiscernible with each other, instead of the similarity class of traditional techniques. As maximal consistent block requires more conditions than similarity class, the granularity of knowledge consisting maximal

consistent block is usually smaller which makes the approximations more precise. But in Deyu Li's paper, the approach of computing maximal consistent block was not provided. So we continued the investigation in this paper.

In this paper, basic notions related to an information system are briefly reviewed in section 2. In section 3, the concept of maximal consistent block and its properties are discussed. Then we gave several approaches of how to compute maximal consistent block in section 4. We conclude the paper with a summary in section 5.

2 Information System

An information system (IS) is an ordered triplet $I = (O, A, f)$, where O is a finite nonempty set of objects and A is a finite nonempty set of attributes. V_a is the domain of an attribute a . Any attribute domain V_a may contain special symbol “*” to indicate that the value of an attribute is unknown. Any domain value different from “*” will be called regular. A system in which values of all attributes for all objects from O are regular (known) is called complete, it is called incomplete otherwise.

Definition 1. ([4][5]) Let $I = (O, A, f)$ be a complete information system. Each subset of attributes $B \subseteq A$ determines a binary indiscernibility relation $IND(B)$ on O :

$$IND(B) = \{(x, y) \in O \times O \mid \forall a \in A, f_a(x) = f_a(y)\}$$

The relation $IND(B)$, $B \subseteq A$ is an equivalence relation and constructs a partition of O . We, however, do not advocate the use of relation $IND(B)$ in an incomplete information system because there are actually situations in which two objects belonging to the same equivalence class of $IND(B)$ may have different properties in reality.

Definition 2. ([4][5]) In an incomplete system $I = (O, A, f)$, a similarity relation is defined as follows:

$$SIM(A) = \{(x, y) \in O \times O \mid \forall a \in A, f_a(x) = f_a(y) \\ \text{or } f_a(x) = * \text{ or } f_a(y) = *\}$$

From the definition of $SIM(A)$, it can be observed that if a pair of objects (x, y) from $O \times O$ is in $SIM(A)$, then they are perceived as similar.

By $S_A(x)$ we denote the set as $\{y \in O \mid (x, y) \in SIM(A)\}$.

3 Maximal Consistent Block

Definition 3. ([1]) Let $I = (O, A, f)$ be an incomplete system, $B \subseteq A$ is a subset of attributes and $X \subseteq O$ is a subset of objects. We say X is consistent with respect to A if $(x, y) \in SIM(B)$ for any $x, y \in X$. If there does not exist a subset $Y \subseteq O$ such that $X \subset Y$, and Y is consistent with respect to B , then X is called maximal consistent block of B .

The concept of a maximal consistent block is adopted from discrete mathematics. It describes the maximal collection of objects in which all objects are similar, i.e., they are

indiscernible in terms of available information provided by B. All maximal consistent blocks determined by $B \subseteq A$ are denoted as $C(B)$, and the set of all maximal consistent blocks of A which includes some object $x \in O$ is denoted as $C_x(B)$.

Example 1. Let $I = (O, A, f)$ be an incomplete information system presented as follows:

Table 1. An incomplete information system

Objects	a ₁	a ₂
x ₁	1	2
x ₂	2	2
x ₃	*	2
x ₄	2	1
x ₅	1	*
x ₆	1	1

The similarity classes determined by A are:

$$\begin{aligned}
 S_A(x_1) &= \{x_1, x_3, x_5\} & S_A(x_2) &= \{x_2, x_3\} \\
 S_A(x_3) &= \{x_1, x_2, x_3\} & S_A(x_4) &= \{x_4\} \\
 S_A(x_5) &= \{x_1, x_3, x_5, x_6\} & S_A(x_6) &= \{x_5, x_6\}
 \end{aligned}$$

The collection of all maximal consistent blocks determined by A is:

$$C(A) = \{Y_1 = \{x_1, x_3, x_5\}, Y_2 = \{x_2, x_3\}, Y_3 = \{x_4\}, Y_4 = \{x_5, x_6\}\}$$

The collections of maximal consistent blocks with respect to (a_1, a_2) are:

$$\begin{aligned}
 C_1(A) &= \{Y_{14}\} & C_2(A) &= \{Y_2\} & C_3(A) &= \{Y_1, Y_{24}\} \\
 C_4(A) &= \{Y_3\} & C_5(A) &= \{Y_1, Y_4\} & C_6(A) &= \{Y_4\}
 \end{aligned}$$

4 Approaches to Compute Maximal Consistent Block

In this section, we first proposed three approaches to compute maximal consistent block.

4.1 Brute Force Method

It is an algorithm based on the concept of the maximal consistent block. Our task is to find all sets that meet the requirement of the definition of maximal consistent block. So we just need to find all these sets step by step. This algorithm can also be regarded as a brute force method.

Here is the main process of this algorithm:

First, we pick a similar pair by comparing any couple of objects. Then we fetch a new object, testing if it's similar to all objects in the pair. If it is, put the object into the pair and the pair become a triplet. If it isn't, test another object. After processing in both condition above, continue testing all other objects, if it's similar to all objects in the

group, put it into them then do the loop again and again until all objects is tested. After this, begin from another pair and repeat the operation above. During the process, when an inner loop is done, record the consistent block to the storage. In order to demonstrate the algorithm, we need to view an example first.

Table 2. An simple incomplete information system

Objects	Attribute value
X_1	1
X_2	2
X_3	*
X_4	2
X_5	1
X_6	3

The pseudocode is presented as follows:

Loop 1. Store X_1 into the memory 1.

A. Compare X_1 with X_2 , turn out not to be similar so jump to B.

B. Compare X_1 with X_3 , turn out to be similar so store X_3 into memory 1. Jump to C.

C. Compare X_1 with X_4 , not similar, jump to D.

D. Compare X_1 with X_5 , similar, then check if X_5 is similar to all objects in memory 1. Jump to a.

a. Compare X_5 with X_3 (X_3 is from memory 1), similar, store X_5 into memory 1, jump to E.

E. Compare X_1 with X_6 , not similar, jump to (2).

Loop 2. Store X_2 into memory 2.

A. Compare X_2 with X_3 , similar, store X_3 into memory 2, jump to B.

B. Compare X_2 with X_4 , similar, jump to a.

a. Compare X_4 with X_3 , similar, store X_4 into memory 2, jump to C.

C. Compare X_2 with X_5 , not similar, jump to D.

D. Compare X_2 with X_6 , not similar, jump to (3).

Loop 3. Store X_3 into memory 3.

A. Compare X_3 with X_4 , similar, store X_4 into memory 3, jump to B.

B. Compare X_3 with X_5 , similar, jump to a.

a. Compare X_5 with X_4 , not similar, jump to C.

C. Compare X_3 with X_6 , similar, jump to a.

a. Compare X_6 with X_4 , not similar, jump to (4).

Loop 4. Store X_4 into memory 4.

A. Compare X_4 with X_5 , not similar, jump to B.

B. Compare X_4 with X_6 , not similar, jump to (5).

Loop 5. Store X_5 into memory 5.

A. Compare X_5 with X_6 , not similar, jump to (6).

Loop 6. Store X_6 into memory 6.

Every loop generates a consistent block.

(1) $\{X_1, X_3, X_5\}$ (2) $\{X_2, X_3, X_4\}$ (3) $\{X_3, X_4\}$

(4) $\{X_2\}$ (5) $\{X_1\}$ (6) $\{X_3\}$

From the result above, we find out some blocks are contained in others, which indicates the results are not maximal consistent blocks but consistent blocks. So we need to merge the result so as to generate maximal consistent block.

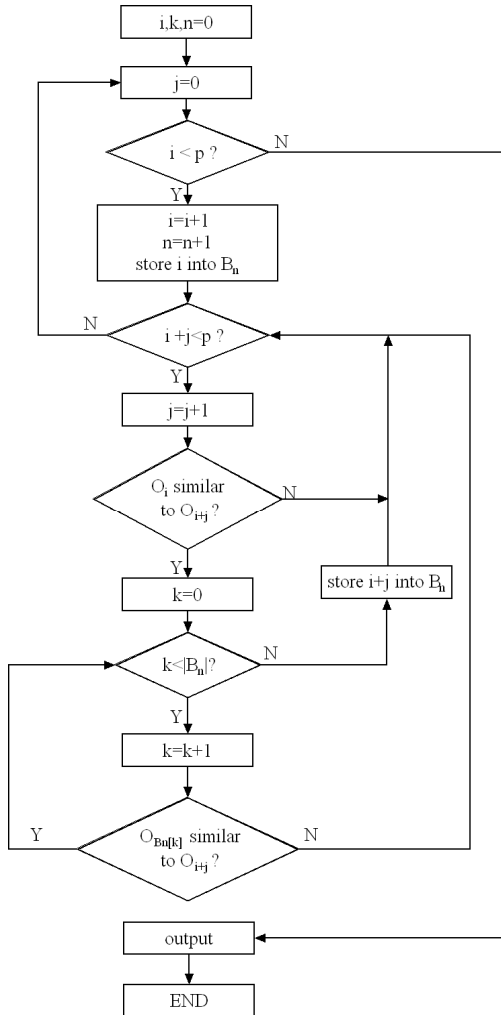


Fig. 1. Flow chart of brute force method

Figure 1 is the detailed flow chart of brute force algorithm. There are several variables with the meaning presented in the following:

i: The algorithm is dealing with the *i*th object, comparing it with all other objects below.

MCB stands for maximal consistent block.

CB stands for consistent block.

j: *j* means the algorithm is comparing the *i*th object with (*i*+*j*)th object.

k: *k* is used to compare the (*i*+*j*)th object with all exiting objects in B_n as a counter.

B_n : B_n is the storage that keeps all objects that meets the requirement of the concept of maximal consistent block. Every time the variable *i* changes, the program generates a new B_n , that is to say *n* is added by one.

$|B_n|$: $|B_n|$ means the amount of objects in B_n .

In this algorithm, an input of an information system is required. We assume there are *p* objects and *r* attributes of the input information system.

It's obvious to see that there are three loops in the flow chart above. First loop will do *p* times as it is shown in flow chat 1, because we have to go through all objects in the system first. We call the object which remains invariant before *i* changes in first loop reference object. The second loop is to check whether all other objects after reference object is similar to reference object. So as the program goes on, the serial number of reference object (in Figure 1 it's referring to *i*) is getting bigger, and the amount of other objects after reference object is getting less. In other words, the second loop' calculation is getting less as program goes on. Here every time *i* is added by 1, the amount of compare in second loop is reduced by 1. With these analysis, the frequency of second loop can be expressed as follows:

$$p + (p - 1) + (p - 2) + \dots + 1 = p \times \frac{p + 1}{2}$$

Notice there is a compare in second loop. The time for this compare is determined by the amount of attributes in the information system. This is a key operation, so we ignore other operation's time such as "*j*=*j*+1". Taken all these into consideration, we have the time that the second loop consumes:

$$r \times p \times \frac{p + 1}{2}$$

The third loop here is used to check a new object whether it's similar to all other objects in corresponding consistent block (here refers to B_n). But the third loop won't run every time after compare in the second loop unless the object is similar to the current reference object. In order to achieve the worst condition, we assume the program will go into the third loop after every compare of second loop. The times that the third loop runs are $|B_n|$, which won't be more than *p*. Also to achieve the worst, we assume every time the program gets into the third loop, $|B_n| = p$. So we could get the times that the third loop runs totally:

$$p \times \frac{p + 1}{2} \times p \times r$$

With these above, we get the running time:

$$t = p + (p \times \frac{p+1}{2} \times r) + (p \times \frac{p+1}{2} \times p \times r)$$

Pick the highest rank of each variable, we have

$$T = O(p^3) \times O(r)$$

which is the time complexity of the proposed algorithm.

4.2 Method Base on Property 1

Property 1. ([1]) Let $I = (O, A, f)$ be an incomplete information system, $B \subseteq A$, $X \subseteq O$, then

$$X \in C(B) \text{ iff } X = \bigcap_{x \in X} S_B(x)$$

The second algorithm is based on property 1. The main idea of this algorithm is to calculate the similar class first, then test them one by one according to property 1. The main work of this algorithm is to compute similar class. So Figure 1 shows the flow chart of this process.

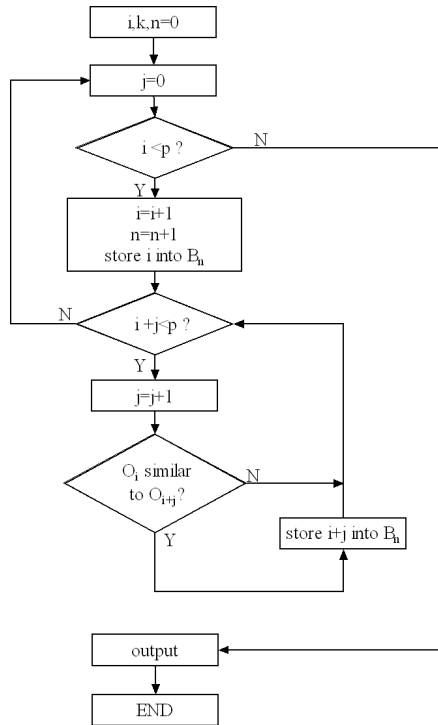


Fig. 2. Flow chart of property 1 method

After we get all similar classes, we need to check all subset of objects in an information system. If an information system has p objects, then the amount of subsets will be:

$$C_p^0 + C_p^1 + C_p^2 + C_p^3 + \dots + C_p^p = 2^p$$

2^p is quite a high rank for an algorithm and this is only a part of this algorithm. So this approach is only available for tiny information systems. With this limitation, there is no need to do further investigation.

4.3 Recursive Method

Property 2. ([1]) Let $I = (O, A, f)$ be an incomplete information system and $B_1 \subseteq B_2 \subseteq A$. For an arbitrary $X \in C(B_2)$ there exists $Y \in C(B_1)$ such that $X \subseteq Y$.

This property shows if you add attributes to the system one by one, the maximal consistent block of original system will be spilt into small pieces. From this point of view, we design the third algorithm.

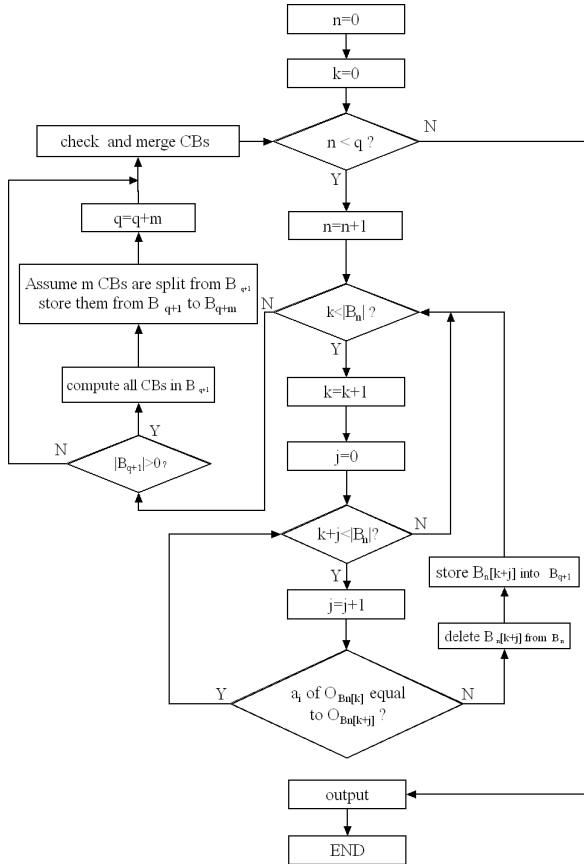


Fig. 3. Flow chart of recursive method

Proposition 1. Let $I = (O, A, f)$ be an incomplete information system and $B_2 = B_1 \cup a, a \in A - B_1, B_1 \subseteq B_2 \subseteq A$. M_i is the consistent block with respect to B_1 while N_j with respect to B_2 . For all N_j , there exist M_i which satisfy $N_j \subseteq M_i$.

It can be regarded as a special case of property 2.

First, pick one attribute of the information system, computing all consistent block with respect to this attribute by using brute force method. Then add one attribute to current attribute set, check these consistent blocks that were calculated before with respect to the latest added attribute. The objects of each consistent block should be checked one by one, testing if it is similar to all other objects. If there is some objects which is not similar with others, delete them from the origin consistent block and gather them to a new set. Perform brute force method on the new set and get more consistent blocks. Store them so we get more consistent blocks. Otherwise, that is to say the consistent block remains consistent with respect to the latest added attribute. In this case, go on to check other blocks until all are checked over.

After all blocks are tested, add one more attribute and test again. The algorithm is done when all attribute is added. Then we get a series of consistent block. Merge them and we'll get the maximal consistent block.

We design the algorithm's flow chart as Figure 3 shows.

In this chart, we also assume the input information system has p objects and r attributes.

q: q represents that there are q consistent blocks with respect to the current attribute set.

n: n means the program is testing the nth consistent block whether it is consistent on new attribute.

k: k is the serial number of object in the nth consistent block. Now the program is testing the kth object of the consistent block whether it is similar to all other objects in the block.

j: j shows the program is checking whether the kth object is similar to the (k+j)th object.

As it is shown in the Figure 3, there are three loops in this algorithm as well. The frequency of first loop depends on q. q is the amount of consistent blocks with respect to current attribute set. It will change as program runs, but will remain less than p. We assume $q=p$ here.

The second loop is used to go over all reference objects. Its frequency is determined by $|B_n|$. Maximal consistent blocks usually have different sizes, so $|B_n|$ will change every time n changes. Let $k_1 = |B_1|, k_2 = |B_2|, \dots, k_q = |B_q|$. It's obvious that $k_n \leq q$. The time this program runs will be:

$$k_1 + k_2 + \dots + k_q \leq q^2$$

The third loop will check all latter objects with the reference object. Every time the loop runs, the objects afterwards become less. So the loop is running smaller and smaller. The frequency of the third loop can be presented as follows:

$$\begin{aligned} & (k_1 + (k_1 - 1) + \dots + 1) + (k_2 + (k_2 - 1) + \dots + 1) + \dots \\ & \quad + (k_q + (k_q - 1) + \dots + 1) \\ & = \frac{k_1 \times (k_1 + 1)}{2} + \frac{k_2 \times (k_2 + 1)}{2} + \dots + \frac{k_q \times (k_q + 1)}{2} \end{aligned}$$

$$= \frac{1}{2}((k_1^2 + k_2^2 + \dots + k_q^2) + (k_1 + k_2 + \dots + k_q)) \leq \frac{1}{2}(p^2 + p^2) = p^2$$
 Now we have the frequency of all loops, but the algorithm is not done yet. It has to run r times in order to add all attributes of an information system. Hence, we get the time complexity of the algorithm.

$$T = O(p^2) \times O(r)$$

Compared with brute force method, this algorithm is more efficient as it costs less time.

5 Conclusion

Maximal consistent block achieves better approximation accuracy than the original approach in [4][5]. In order to make it into use, we proposed three approaches to compute the maximal consistent block. The first approach is the most direct and it is also easy to use. The second approach is just a rudiment because of its high complexity of time. We presented this method here, hoping to provide a direction that will help the algorithm reduce its complexity with more work by others. Afterwards, we further investigated the time complexity of these approaches. It turns out that the third one is the best of the three with the least time complexity. All these approaches have been designed into a flow chart, which can be transformed into a program language quickly. With these methods, the maximal consistent block technique for rule acquisition will be more and more widely used. And the approaches to compute maximal consistent block will be improved reciprocally as a result.

Acknowledgements. This work was also supported by the National Natural Science Foundation of China (Nos.60963006, 61173181), the Humanities and Social Science funds Project of Ministry of Education of China (No. 09YJCZH082, 11XJJAZH001), and the Science and Technology Project of Qingdao (No.12-1-4-4-(9)-jch).

References

1. Leung, Y., Li, D.: Maximal consistent block technique for rule acquisition in incomplete systems. *Information Sciences* **153**, 85–106 (2003)
2. Dai, J., Qing, X.: Approximations and uncertainty measures in incomplete information systems. *Information Sciences* **198**, 62–80 (2012)
3. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Sciences* **11**, 341–356 (1982)
4. Kryszkiewicz, M.: Rough set approach to incomplete information systems. *Information Sciences* **112**, 39–49 (1998)
5. Kryszkiewicz, M.: Rules in incomplete information systems. *Information Sciences* **113**, 271–292 (1999)
6. Leung, Y., Wu, W.-Z., Zhang, W.-X.: Knowledge acquisition in incomplete information systems: A rough set approach. *European Journal of Operational Research* **168**, 164–180 (2006)

7. Xu, W., Li, Y., Liao, X.: Approaches to attribute reductions on rough set and matrix computation in inconsistent ordered information systems. *Knowledge-Based Systems* **27**, 78–91 (2012)
8. Shao, M.W., Zhang, W.X.: Dominance relation and relus in an incomplete ordered information system. *Int. J. Intell. Syst.* **20**, 13–27 (2005)
9. Wei-hua, X., Xiao-yan, Z., Wen-xiu, Z.: Knowledge granulation, knowledge entropy and knowledge uncertainty measure in ordered information systems. *Applied Soft Computing* **9**, 1244–1251 (2009)
10. Zhai, L.-Y., Khoo, L.-P., Zhong, Z.-W.: A dominance-based rough set approach to Kansei Engineering in product development. *Expert Systems with Applications* **36**, 393–402 (2009)
11. Leuang, Y., Wu, W.Z., Zhang, W.X.: Knowledge acquisition in incomplete information systems: a rough set approach. *European Journal of Operational Research* **168**(1), 164–180 (2006)
12. Peters, G., Poon, S.: Analyzing IT business values – A Dominance based Rough Sets Approach perspective. *Expert Systems with Applications* **38**, 11120–11128 (2011)
13. Fan, T.-F., Liau, C.-J., Liu, D.-R.: Dominance-based fuzzy rough set analysis of uncertain and possibilistic data tables. *International Journal on Approximate Reasoning* **52**, 1283–1297 (2011)
14. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based Rough Set Approach to decision under uncertainty and time preference. *Ann. Oper. Res.* **176**, 41–75 (2010)
15. Huang, B., Li, H.-X., Wei, D.-K.: Dominance-based rough set model in intuitionistic fuzzy information systems. *Knowledge-Based Systems* **28**, 115–123 (2012)
16. Yang, X., Zhang, M., Dou, H., Yang, J.: Neighborhood systems-based rough sets in incomplete information system. *Knowledge-Based Systems* **24**, 858–867 (2011)
17. Liou, J.J.H., Tzeng, G.-H.: A Dominance-based Rough Set Approach to customer behavior in the airline market. *Information Sciences* **180**, 2230–2238 (2010)
18. Yang, X., Yu, D., Yang, J., Wei, L.: Dominance-based rough set approach to incomplete interval-valued information system. *Data and Knowledge Engineering* **68**, 1331–1347 (2009)
19. Iyer, N.S.: A family of dominance rules for multiattribute decision making under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics Part A* **33**, 441–450 (2003)
20. Xu, Z.S.: *Uncertain Multiple Attribute Decision Making: Methods and Applications*. TsingHua Press, Beijing, China (2004)
21. Yang, X., Yu, D., Yang, J., Wei, L.: Dominance-based rough set approach to incomplete interval-valued information system. *Data and Knowledge Engineering* **68**, 1331–1347 (2009)
22. Qian, Y.H., Liang, J.Y., Dang, C.Y.: Interval ordered information systems. *Computers and Mathematics with Applications* **56**, 1994–2009 (2008)

Missing Value Imputation for the Analysis of Incomplete Traffic Accident Data

Rupam Deb^(✉) and Alan Wee-chung Liew

School of Information and Communication Technology, Griffith University,
Logan, Australia
rupam.deb@griffithuni.edu.au,
a.liew@griffith.edu.au

Abstract. Road traffic accidents are a major public health concern, resulting in an estimated 1.3 million deaths and 52 million injuries worldwide each year. All the developed and developing countries suffer from the consequences of increase in both human and vehicle population. Therefore, methods to reduce accident severity are of great interest to traffic agencies and the public at large. To analysis the traffic accident factors effectively we need a complete traffic accident historical database without missing data. Road accident fatality rate depends on many factors and it is a very challenging task to investigate the dependencies between the attributes because of the many environmental and road accident factors. Any missing data in the database could obscure the discovery of important factors and lead to invalid conclusions. In order to make the traffic accident datasets useful for analysis, it should be preprocessed properly. In this paper, we present a novel method based on decision tree and imputed value sampling based on correlation measure for the imputation of missing values to improve the quality of the traffic accident data. We applied our algorithm to the publicly available large traffic accident database of United States (explore.data.gov), which is the largest open federal database in United States. We compare our algorithm with three existing imputation methods using three evaluation criteria, i.e. mean absolute error, coefficient of determination and root mean square error. Our results indicate that the proposed method performs significantly better than the three existing algorithms.

Keywords: Data mining · Data preprocessing · Decision tree · Data imputation · Traffic accident

1 Introduction

The high growth of the number of vehicles leads to roads with higher traffic density. The immediate effect of this situation is the dramatic increase of traffic accidents on the road, which has become a serious problem in many countries. For example, 2478 people died on Spanish roads in 2010, which means one death for every 18,551 inhabitants [1]. In United States (according to Department of Transportation, United states) nearly 30,000 people died in road accidents. According to Australian Bureau of

Statistics, the majority of transport related deaths (almost 72%) in Australia is associated with motor vehicles driven on public roads.

In recent years, there has been rapid development in sensor technologies. As a result, huge amount of traffic accident data has been collected [2]. Due to the wide availability of data and the imminent need for turning such data into useful information and knowledge, data mining has attracted a great deal of attention [3]. Using data mining technology such as classification and clustering, we can uncover patterns of traffic activities and factors that lead to accident.

To run the classification and clustering algorithms, there is a strong need for data preprocessing to ensure the data is of good quality. Data preprocessing takes almost 80% of the total data mining effort. It is also known that good results can be achieved by using data mining algorithms only if one has a good quality dataset [4].

In general, data preprocessing includes imputation of missing values, smoothing out noisy data, identification of incorrect data, and correction of inconsistent data. In this paper, we propose a new decision tree-based algorithm called DSMI for imputing missing values. Our experiments show that the proposed algorithm has better imputation accuracy compared with accuracy produced by several other existing algorithms.

The paper is organized as follows: In section 2 we present a literature review of related work. Our proposed algorithm is described in section 3. Experimental results are discussed in section 4. Finally section 5 draws the concluding remarks.

2 Related Work

Many missing value imputation approaches have been proposed recently for various applications [5-8, 10-13]. Among them, k -Decision tree based imputation (k DMI) [5], Decision tree based imputation (DMI) [6], Expectation Maximization imputation (EMI) [7], and k -Nearest Neighbor based Imputation (kNNI) [8] are some well-known imputation methods.

For imputing numerical missing values, EMI algorithm relies on estimating the mean and covariance matrices of the dataset. The EMI algorithm starts with initial estimates of the mean and the covariance matrix and cycles through the steps until the imputed values and the estimates of mean and covariance matrix stop changing appreciably from current iteration to the next iteration [7]. The main drawback of this method is that for imputing the missing value it uses information from the whole dataset and therefore is suitable only for datasets that exhibits strong correlations for the attributes within the whole dataset.

kNNI method first finds user-defined k number of records from the total dataset by using the Euclidean distance measure. For imputing a numerical missing value the method utilizes the mean value of the specific attribute within the k most similar records of the entire dataset. If the missing attribute is categorical then the method utilizes the most frequent value of the attribute within the k most similar records. kNNI is a simple method that performs well on the dataset having strong local correlation structure. However, the method can be expensive for a large dataset since for each record

having missing value(s) it finds k number of similar records by searching the whole dataset. Moreover, the identification of a suitable value for k can be a challenging task [8].

Rahman *et al.* proposed the DMI [6] technique which uses the decision tree and EM algorithm for missing value imputation. They argued that the correlations among attributes within a horizontal partition of a dataset can be higher than the correlations over the whole dataset. This technique works as follows: it first divides the full dataset (D_{Full}) into two sub datasets, one having records with missing values (D_{Miss}) and others having records without missing values ($D_{Complete}$). Then it builds decision trees on $D_{Complete}$ considering the attributes having missing values in D_{Miss} as class attributes. After that, it assigns each record with missing value(s) in D_{Miss} to the leaf where it falls in for the tree that considers the attribute, which has a missing value for the record, as the class attribute. Finally, it imputes numerical missing values using EM algorithm or categorical missing values using majority class values within the leaves. They showed that DMI performed well compared with other existing imputation methods. However, for imputing categorical values, simple voting is used. Another more serious problem is that the authors did not define how the imputation is done if the missing values record falls in more than one leaves, a situation that could occur if there is more than one missing values in a record.

k DMI [5] algorithm imputes missing values using two levels partitioning. Like DMI, k DMI algorithm also employed horizontal partitioning based on a decision tree in first level partitioning. For second level partitioning, the authors used a BestKNN approach to first find the best value of k by searching all records of a leaf and calculated the root mean square error (RMSE) of the non-missing attribute values. Then EM algorithm is used for imputing numerical data and frequent value of BestKNN is used for imputing categorical data, respectively. However, it is not clear if all the attributes of a record are categorical then how would RMSE being calculated using BestKNN. It is also not clear if the missing values record falls in more than one leaves, how the imputation would be done. Here, we propose DSMI, a new decision tree with random sampling of imputed values imputation method to address the above problems. The justification for using decision tree-based method is that the records in a leaf are more correlated than the whole dataset and the variance between attribute values for the records within a leaf is smaller than that of the entire dataset [9].

3 Proposed Approach

We present our missing values imputation technique below.

We build the decision trees with correlated records using C4.5 algorithm. The records with missing values are assigned to the leaves and their missing values are imputed using correlation measure.

3.1 Basic Concept

We illustrate our basic algorithm design here. At first, the full dataset (D_{Full}) is divided into two sub datasets. One subset contains records with missing values (D_{Miss}) and other one without missing values ($D_{Complete}$). A sample datasets D_{Full} , $D_{Complete}$, and D_{Miss} are shown in TABLE 1, 2, and 3. Then, we build a set of decision trees on $D_{Complete}$ with the attributes having missing values in D_{Miss} , as the class attributes. For example in D_{Miss} three attributes (Driver status, Passenger number and Accident address) have missing values and we make three decision trees depend on these attributes shown in Figure 1, 2, and 3.

Table 1. Full dataset d_{full}

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3	Sanders	Kill
R2	Drunk	Good	4	?	Kill
R3	Drunk	Good	2	Glendale	No injury
R4	Normal	Fair	3	Glendale	No injury
R5	Normal	Fair	?	Glendale	No injury
R6	?	Good	?	Glendale	Kill

Table 2. Complete dataset $d_{complete}$

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3	Sanders	Kill
R3	Drunk	Good	2	Glendale	No injury
R4	Normal	Fair	3	Glendale	No injury

Table 3. Miss dataset d_{miss}

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R2	Drunk	Good	4	?	Kill
R5	Normal	Fair	?	Glendale	No injury
R6	?	Good	?	Glendale	Kill

In Figure 1, ‘ Leaf 1 Sanders: (1) R1’ represents 1 record associated with ‘Sanders’ value and ‘R1’ record falls in ‘Leaf 1’. Here, we quantize the numerical attribute by the square root of its domain size. As passenger attribute is numerical so we have to quantize it and it is shown in Table 4.

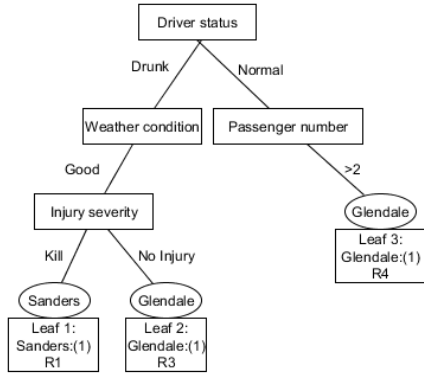


Fig. 1. Tree for accident address

Table 4. Complete dataset for numeric category passenger number

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3-4	Sanders	Kill
R3	Drunk	Good	1-2	Glendale	No in-jury
R4	Normal	Fair	3-4	Glendale	No in-jury

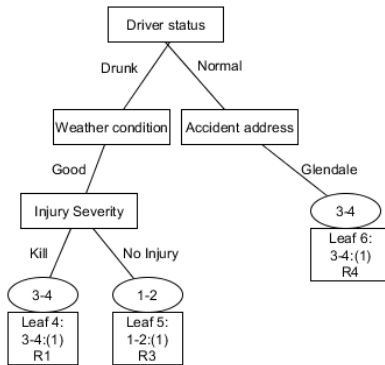


Fig. 2. Tree for passenger number

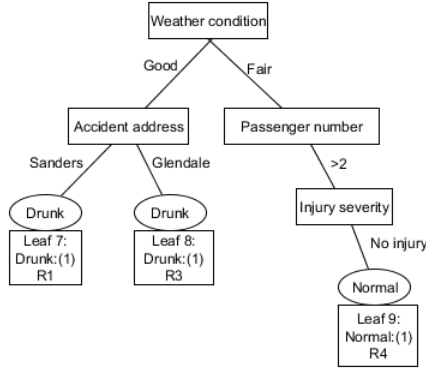


Fig. 3. Tree for driver status

After that, we assign each record of D_{Miss} to the leaf of a tree with the same attribute as the missing attribute. For example, in TABLE 3, R2 record has ‘Accident address’ attribute value missing so we assign this record to the Accident address tree’s (Figure 1) leaf. R2 and R5 records are assigned to Leaf 1 and Leaf 6, respectively which are shown in TABLE 5 and TABLE 6.

Table 5. Assign miss dataset for accident address

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3	Sanders	Kill
R2	Drunk	Good	4	?	Kill

Table 6. Assign miss dataset for passenger number

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R4	Normal	Fair	3	Glendale	No injury
R5	Normal	Fair	?	Glendale	No injury

The records with more than one missing values would fall into multiple leaves. If any record falls in multiple leaves, we aggregate records in all these leaves into one collection. According to Table 3, R6 record has two missing values so it is assigned to three leaves 4, 6, and 8 (Figure 2 and 3) and shown in Table 7, 8, and 9. Then, we aggregate these three tables into Table 10.

Table 7. Assign miss dataset for passenger number and driver status

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3	Sanders	Kill
R6	?	Good	?	Glendale	Kill

Table 8. Assign miss dataset for passenger number and driver status

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R4	Normal	Fair	3	Glendale	No injury
R6	?	Good	?	Glendale	Kill

Table 9. Assign miss dataset for passenger number and driver status

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R3	Drunk	Good	2	Glendale	No injury
R6	?	Good	?	Glendale	Kill

Table 10. Combined table 7, 8, and 9

	<i>Driver status</i>	<i>Weather condition</i>	<i>Passenger number</i>	<i>Accident address</i>	<i>Injury severity</i>
R1	Drunk	Good	3	Sanders	Kill
R4	Normal	Fair	3	Glendale	No injury
R3	Drunk	Good	2	Glendale	No injury
R6	?	Good	?	Glendale	Kill

We impute the missing values in each table by searching for records in the table, which have the maximum number of non-missing attributes in common to the missing record. Then the attribute values corresponding to the missing values in the selected records are taken to be possible imputed values. For example, in TABLE 10, R6 record has three non-missing values but we do not get any record matching with these three non-missing values. So we search instead for two matching non-missing values and get two records with two matching attributes values: R1(Good, Kill) and R3(Good, Glendale). For the two missing attributes (Driver status, Passenger number), the possible imputed values from R1 and R3 records are (Drunk, 3), and (Drunk, 2), respectively.

If there is more than one possible imputed value, the correlation between each possible imputed value and the matched non-missing value is computed using IS measure. The IS measure between two items a and b is given by $IS(a,b) = P(a,b)/\text{Sqrt}(P(a)*P(b))$, where P denotes probability value [14]. For example, in Table 10, the IS measure of (Drunk, 3) and (Drunk, 2) have the same value of 1. Finally, the actual imputed value is obtained by random sampling from the list of possible imputed values based on the

distribution of their IS measures. For example, since both (Drunk, 3) and (Drunk, 2) have the same IS measure of 1, their distribution is both equal to 0.5, and both have equal probability of been chosen as the actual imputed values for the missing values in R6. Random sampling according to the IS measure ensures that uncertainty in attribute values are modeled for and helps to reduce bias artifact in the imputed dataset.

3.2 Proposed Algorithm

Our proposed DSMI algorithm is presented below.

<p>Input: Full dataset with missing values Output: Full dataset with missing values imputed</p>
<p>D_{Full}: Full dataset $D_{Complete}$: Sub dataset without missing values D_{Miss}: Sub dataset with missing values M: Total number of attributes having missing values in D_{Miss} A_i: i-th attribute in dataset L_j: j-th leaf created from the attribute A_i, which has missing value(s) in D_{Miss} N: Total number of non-missing records matched with a missing record in table T O_k: possible imputed value(s) from the k-th record of T C_k: k-th record IS measure value corresponds to O_k</p>
<p>Step I: Decompose full dataset into complete and missing values sub datasets $D_{Full} = D_{Complete} + D_{Miss}$ Step II: Generate decision trees with leaves FOR i = 1 to M IF A_i is numeric attribute then Quantize attribute using $\sqrt{ A_i }$ where A_i is the domain size of A_i END IF Create decision tree with leaves using C4.5 from $D_{Complete}$ considering A_i as class attributes which have missing values in D_{Miss} END FOR Step III: Assign each record of D_{Miss} into leaf of the missing attribute(s) tree(s) and create table of related records FOR each record of D_{Miss} DO Get missing attribute(s) associated with this record Assign record into the corresponding leaf L_j Generate table T from records in L_j END FOR IF a record is assigned into multiple leaves THEN</p>

```

Aggregate all tables into one table T
Remove duplicate record(s) from this table
END IF
Step IV: Impute missing values
FOR each table T DO
  FOR each missing record in T DO
    Find non-missing records in T that match with the
    maximum number of non-missing attribute(s) in the
    missing record, and let N be the number of such records
    FOR k = 1 to N
       $O_k$  = possible imputed value(s) from the k-th matched
      record
       $C_k$  = IS measure computed for  $O_k$ 
    END FOR
    Imputed value(s) is obtained by randomly samples from
    the set of possible imputed values {  $O_1 \dots O_N$  } based on the
    distribution specified by the set of IS measures {  $C_1 \dots C_N$  }
  END FOR
END FOR
Step V: Get full dataset with missing values imputed

```

4 Results and Discussion

We compare performance of the method with three imputation methods *k*DMI, DMI, and EMI. Imputation accuracy is evaluated using three well-known performance indicators: mean absolute error (MAE), coefficient of determination (CoD), and root mean square error (RMSE). CoD is used in statistical model analysis to assess how well a model predicts future outcomes. The RMSE is a frequently used measure of the difference between values predicted by a model and the values actually observed. MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. The higher the CoD, the lower the RMSE and MAE, the better the imputation.

We do experiment on 22 text files data (Large Truck Crash Causation Study File 1). The text files have different number of attributes and 60639 records. Most of the attributes (90%) are categorical and coded with numeric values which correspond to text descriptions. The dataset contains 2193 records with missing values. Here, we first remove the records having missing values thereby create a dataset having 58446 records without any missing values.

We use four types of missing patterns [6]. In simple pattern a record can have at most one missing value. In medium pattern, a record can have missing values for 2 – 50 % of the total number of attributes. In a complex pattern, a record can have missing values for 51 – 80 % of the total number of attributes. A blended pattern contains 25% records having missing values with simple pattern, 50% with medium pattern and 25% with

complex pattern. We also use two types of missing models, namely overall and uniformly distributed (UD). In the UD missing model, each attribute has equal number of missing values. However, in the overall model, missing values are not equally distributed among attributes.

Here, we artificially create missing values in the dataset by using 4 missing patterns, namely simple, medium, complex and blended, 4 missing ratios i.e. 2%, 4%, 8% and 12%, and two missing models, namely overall and uniformly distributed (UD). We have altogether 32 missing combinations (4 missing ratios, 4 missing patterns, 2 missing models). For each combination we use 200 datasets. So, we create 6400 (32 combinations, 200 datasets for each combination) datasets.

Figure 4 shows that DSMI performed well for imputing categorical missing values compare with others algorithm. This performance is very important for preprocessing our datasets as most attributes of traffic accident dataset are categorical.

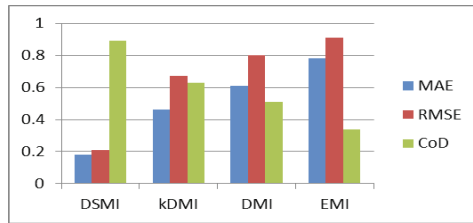


Fig. 4. Categorical missing value imputation

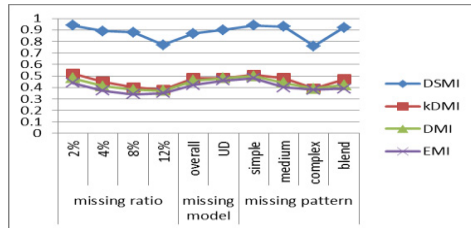


Fig. 5. Average performance based on coefficient of determination (CoD)

We present the aggregate performances based on CoD for missing ratio, missing model and missing patterns on the traffic accident datasets in Figure 5. The figure shows that for all cases DSMI outperforms other methods in terms of CoD. The result of missing values imputation for numerical values is presents in Figure 6. In Figure 7, we present the aggregate performances based on RMSE for 2 missing ratios, 2 missing models and 4 missing patterns. Our algorithm achieves the best performance with all performance indicators. In practical applications, almost all historical databases are categorical. So, we need an algorithm which performs well with categorical missing values. Our results show that the proposed algorithm performed well in all cases.

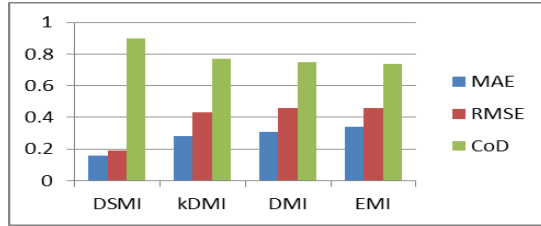


Fig. 6. Numerical missing value imputation

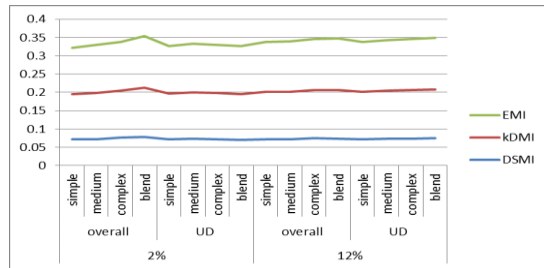


Fig. 7. Average performance based on RMSE

5 Conclusion

In this paper, we proposed a new imputation method with the aim of analyzing traffic accidents data. Our algorithm combines features of decision tree and imputed value sampling based on correlation measure, and has been shown to outperform several popular imputation methods on traffic accident data, where a large number of attributes are categorical.

References

1. Fogue, M., Garrido, P., Martinez, F.J., Cano, J.-C., Calafte, C.T.: A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms. *Expert Systems with Applications* **40**(1), 323–336 (2013)
2. Zamani, Z., Poumand, M., Saraee, M.H.: Application of data mining in traffic management: Case of city of Isfaha. In: *Proceeding of ICECT2010 Conference*, Kuala Lumpur, pp. 102–106 (May 2010)
3. Shanthi, S., Ramani, R.G.: Feature relevance analysis and classification of road traffic accident data through data mining techniques. In: *Proceeding of WCECSC2012 Conference*, San Francisco (October 2012)
4. Miksovsky, P., Matousek, K., Kouba, Z.: Data pre-processing support for data mining. In: *Proceeding of IEEE SMC2002 Conference*, Hammamet, pp. 1–8 (October 2002)

5. Rahman, M.G., Islam, M.Z.: k -DMI: A novel method for missing values imputation using two levels of horizontal partitioning in a data set. In: Proceeding of ADMA2013 Conference, Hangzhou, pp. 250–263 (December 2013)
6. Rahman, M.G., Islam, M.Z.: A decision tree-based missing value imputation technique for data pre-processing. Proceeding of AusDM2011 Conference, Ballarat, pp. 41–50 (December 2011)
7. Schneider, T.: Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate* **14**(5), 853–871 (2001)
8. Batista, G.E.A.P.A., Monard, M.C.: An analysis of four missing data treatment methods for supervised learning. *Journal of Applied Artificial Intelligence* **17**(8(5-6)), 519–533 (2003)
9. Islam, M.Z., Brankovic, L.: Privacy preserving data mining: A noise addition framework using a novel clustering technique. *Knowledge-Based systems* **24**(8), 1214–1223 (2011)
10. Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, T., Kolehmainen, M.: Methods for imputation of missing values in air quality data sets. *Journal of Atmospheric Environment* **38**(18), 2895–2907 (2004)
11. Liew, A.W.C., Law, N.F., Yan, H.: Missing value imputation for gene expression data: computational techniques to recover missing data from available information. *Briefings in Bioinformatics* **12**(5), 498–513 (2011)
12. Gan, X., Liew, A.W.C., Yan, H.: Microarray missing data imputation based on a set theoretic framework and biological consideration. *Nucleic Acids Research* **34**(5), 1608–1619 (2006)
13. Cheng, K.O., Law, N.F., Siu, W.C.: Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data. *Pattern Recognition* **45**(4), 1281–1289 (2012)
14. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. *Information Systems* **29**(4), 293–313 (2004)

Learning Behaviour for Service Personalisation and Adaptation

Liming Chen¹(✉), Kerry Skillen², William Burns², Susan Quinn², Joseph Rafferty²,
Chris Nugent², Mark Donnelly², and Ivar Solheim³

¹ School of Computer Science and Informatics, De Montfort University, Leicester, UK
liming.chen@dmu.ac.uk

² School of Computing and Mathematics, University of Ulster, Ulster, UK
{k.skillen, wp.burns, cd.nugent, mp.donnelly}@ulster.ac.uk,
{Quinn-S47, Rafferty-J}@email.ulster.ac.uk

³ Norwegian Computing Center, Oslo, Norway
ivar.solheim@nr.no

Abstract. Context-aware applications within pervasive environments are increasingly being developed as services and deployed in the cloud. As such these services are increasingly required to be adaptive to individual users to meet their specific needs or to reflect the changes of their behavior. To address this emerging challenge this paper introduces a service-oriented personalisation framework for service personalisation with special emphasis being placed on behavior learning for user model and service function adaptation. The paper describes the system architecture and the underlying methods and technologies including modelling and reasoning, behavior analysis and a personalisation mechanism. The approach has been implemented in a service-oriented prototype system, and evaluated in a typical scenario of providing personalised travel assistance for the elderly using the help-on-demand services deployed on smartphone.

Keywords: Personalisation · Behavior learning · Adaptation · Pervasive systems · Semantic modeling · Assistive living

1 Introduction

Recently, two trends have emerged for context-aware applications in pervasive environments. Firstly, context-aware applications are increasingly being developed as services, which are deployed in the cloud so that they can be shared and reused by large user cohorts in the same application contexts. Secondly, the rapid development of mobile computing technologies in conjunction with the upsurge of smartphone users has driven the application domains from close-world smart environments towards open-world changing environments. In such application scenarios, smartphones provide mobile users with front-end interfaces to interact with embedded sensors and devices, and to deliver functions in changing environments. These two trends have subsequently led to three emerging research questions, namely (i) how the needs of

different users are met by the same application, (ii) how the varying needs of one user in different environments are met as they transition from one environment to another, and (iii) how the changing needs of a user are dealt with as and when they become familiar with the application scenarios and develop sufficient knowledge and capabilities.

To address the aforementioned issues context-aware, personalised and just-in-time service provisioning will be key. While significant progress has been made in context awareness [7], it still remains a challenge for services to provide the 'right' information for the 'right' user at the 'right' time in the 'right' way. To achieve this, context-awareness alone is insufficient. Personalisation is therefore required to tailor services based on users' unique preferences, needs or capabilities. Comparing to personalisation in web information retrieval [1] and user interface design [2], service personalisation for mobile users in pervasive environments poses a number of challenges. Firstly, a user within different environments typically has varying behaviors. The required services will change dynamically as a user moves around. For example, in a home environment people will usually carry out typical activities related to daily living, while those who travel will have different activities relating to their location or information seeking. Secondly, in a specific environment there may be multiple services each providing a specific service or targeted towards a specific cohort of users. For example, in a smart care home, services could be provided for dementia patients, diabetes patients or the elderly, and each patient may have their own activity profiles. Thirdly, even in the same environment a user's behavior changes over time to reflect new situations or experiences. As a result, the same person may require different levels of services. Fourthly, in web information retrieval or interface design, the application context is relatively static but in changing pervasive environments the surroundings and events within the environments, e.g. in a train station or shopping mall, and also the network connectivity, e.g. communication bandwidth are all dynamic which not only impact on the required services but also the way such services are delivered.

To address these unique characteristics of pervasive context-aware applications as described above, this paper contributes to knowledge in three aspects. Firstly, it introduces a hybrid approach to create and adapt user models based on user behaviours. Secondly, it develops usage data mining methods to learn a user's changing behavior and extract longitudinal patterns. Thirdly, it introduces a service-oriented personalisation framework enabling personalisation services for service personalization.

The remainder of the paper is organised as follows: Section 2 discusses existing related work and highlights the key knowledge contributions. Section 3 introduces the overall conceptual system architecture followed by a detailed description of underlying technologies for the main components, in Section 4. Section 5 outlines a case study including the details of the implementation, testing and evaluation. Section 6 concludes the paper with a brief discussion on future work.

2 Related Work

Personalisation has been widely studied in web information retrieval 1 and interface design 2. In user modelling there are two main approaches. One is to apply statistic or probability analysis methods to perform data mining to extract user models from user behavioural data 3. The other is the knowledge-based user modelling approach which makes use of knowledge engineering techniques, e.g. formal knowledge acquisition, modelling and representation, to build user models. This approach, in particular ontological user modelling, has attracted increasing attention for user modelling of context-aware applications due to their interoperability and ability for knowledge sharing and reuse across several application domains, e.g. COBRA-ONT 4, OntobUM 5 and UPOS 6. Regarding personalisation mechanisms there are currently three broad categories of techniques, namely case-based personalisation 7, collaborative filtering 6 and rule-based reasoning 8. Existing approaches to personalisation have three main drawbacks. Firstly, current user models are normally created as a one-off activity leading to fixed user models. It is usually the case that user behaviours change over time, e.g. previously required help is not needed as the users gain knowledge, experience and capabilities. In this case previous user models will not reflect the exact needs of users. Secondly, current user modelling and personalisation usually target one application scenario with relatively stable application contexts, which did not take into consideration the changing service needs and application context. Thirdly, current personalisation functions have been mainly implemented implicitly in a stand-alone manner. It is difficult for them to be shared and reused by distributed pervasive applications which adopt more and more smartphone frontends. To date, little work has been undertaken to fully address these challenges which have originated from the unique characteristics of service personalisation of pervasive context-aware applications for mobile users.

Consequently, the novelties of the proposed approach are three fold. Firstly, user modelling is not a one-off endeavor, however, more an iterative process which will start by developing ontological user models through knowledge engineering techniques and then enhance the models incrementally and dynamically through behaviour learning. Secondly, personalisation will take into consideration the application context and changing environments, e.g. network connectivity, which are unique to pervasive computing in mobile environments. In addition, the proposed approach will be implemented in a service-oriented architecture, i.e. all user models and personalisation mechanisms will be wrapped as personalisation services, thus making it applicable to wider pervasive context-aware applications.

3 The System Architecture

Figure 1 presents the system architecture of personalisation services for service personalisation within context aware pervasive systems. The architecture aims to enable and support personalisation features of multiple pervasive applications for multiple users in a diversity of pervasive environments, using smartphone frontends.

As can be seen in the client side of the architecture there exist many pervasive applications in intelligent pervasive environments such as smart homes, smart hospitals, conferences in addition to open-world use scenarios such as travelling and shopping. For each application there are many users and each may have different needs and desires. To address the diversity of applications and the heterogeneity of users, traditional approaches to personalisation will be not effective. Rather than providing a user model and personalisation mechanism for each application and for each user, it will be more scalable and realistic to provide personalisation features as services. In this case, context information of an application and the interaction usage data between a user and corresponding applications will be monitored through various sensors from the pervasive environment and the application frontend interfaces. The collected behaviour and usage data will be stored locally and transferred to remote servers for advanced processing. Based on the results of data analysis, users' behavior and usage patterns can be extracted which can in turn be used to update user models or directly inform service adaptation either in terms of the content or interfaces.

The services in the server-side of the architecture interact with each other and also the client-side functions to provide enabling technologies for personalisation and adaptation of application services. The User Profile Services contain user models of various users and a repository of instantiated user profiles which are supposed to be used for personalisation of different pervasive applications. Initial user models will be created through ontological engineering as user ontologies, which will be later updated based on a user's behaviour analysis in the Data Analysis Services. The Usage Data Services consists of the usage data models of an application and its recorded service usage data, including the configuration of the application interfaces. Usage data will be collected from the application frontends in the client side, stored locally and transferred to a remote server in an asynchronous manner, e.g. in order to save power or use better communication bandwidth. The Context Services will be composed of contextual information systems and a context analysis module. The former will record and store low level contextual information collected from sensor data in the environments of pervasive applications, and the latter will analyse the low level contextual data to extract high level meaningful context in terms of application characteristics and scenarios. All User Profile Services, Usage Data Services and Context Services will support recording, storage, retrieval, query and search functions.

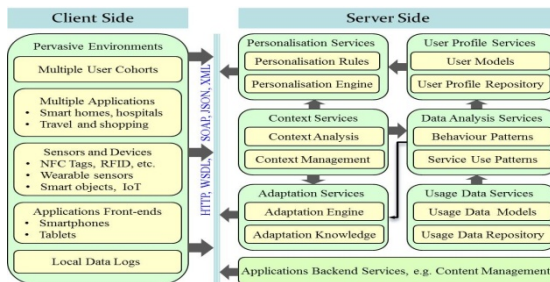


Fig. 1. The system architecture of personalisation services

Central to the architecture are the Data Analysis Services, Personalisation Services and Adaptation Services. Data Analysis Services play two roles. One is to analyse the user's behaviour based on high level context, such as daily activities in a smart home environment. Behaviour analysis will be able to extract a user's behaviour pattern and identify the change of a user's behaviour. This will, in turn, be used to update user models in the User Profile Services. The second role is to analyse the usage data of a service, e.g. which functions have been used in which context. The analysis will extract service usage patterns, e.g. the most frequently used feature(s), the volume or font size of the adopted user interfaces. Such findings will then be used to customise the manner in which a service is delivered. The Personalisation Services consists of domain-dependent, application-specific personalisation knowledge and a reasoning engine. The personalisation knowledge, e.g. rules, will be captured and modelled based on application characteristics and scenarios. The engine will take as inputs user models, user application requests and the context, and then perform reasoning over the personalisation rules. It will then inform applications, e.g. backend services such as a content provision system, of the personalised ways of a service being delivered. Similar to Personalisation Services, Adaptation Services contain application specific adaptation rules and an adaptation reasoning engine. Adaptation will mainly address issues relating to applications in changing environments. As such, it will take inputs based on the application context and/or user application requests, and will then perform reasoning over the adaptation rules. It will subsequently adapt the manner in which a service is delivered based on the situation. For example, consider a content service that is configured to provide a video based instruction to a user, based on the user's personal profile, i.e. the user prefers video. Nevertheless, the system detects that the available internet connectivity is poor and as a result the adaptation service adapts the delivery of the content from video to audio or text.

4 User Modelling, Personalisation and Behavior Learning

The presented approach and the service-oriented architecture has been considered in the EU funded MobileSage project¹. MobileSage intended to develop help-on-demand (HoD) services for elderly people, which allow them to carry out and solve everyday tasks and problems in the self-serve society when and where they occur, just-in-time. A typical example scenario is that a user travels within a foreign country and requires assistance to operate a ticket machine to buy a ticket, e.g. in an airport or train station, without understanding the local language. The hypothesis is that the ticket machine is a smart object equipped with Near Field Communication (NFC) or Quick Response (QR) code technologies 9. When users use the HoD services, deployed in their smartphones (as an app) to interact with the smart ticket machine, the HoD service will retrieve instructions on how to buy a ticket from a content management system in the cloud. The instructions could be delivered in different modalities, e.g. video, audio or text, or presented in different interfaces, e.g. layout, volume or font size, based on the user's profile, e.g. preference and capabilities, and the situation of the point of help. The

¹ The MobileSage project <http://www.mobilesage.eu/>

following describes the underlying technical details of the personalisation services and Section 5 outlines the implementation details of the overall system.

4.1 User Profile Modelling and Personalisation Mechanism

Based on MobileSage application scenarios, three cohorts of end users had been considered in terms of their education background, medical conditions and cognitive capabilities. The user modelling tool PERSONAS was then used to characterise the main concepts and relationships of each cohort which leads to the conceptual user model (refer to Figure 2). To accommodate the diversity of pervasive applications in different environments, the conceptual user models have covered a wide range of entities and relationships. While conceptual models can be serialised in any format, we have used the ontological engineering tool Protégé to develop computational user models, which are subsequently represented in OWL (refer to Figure 3).

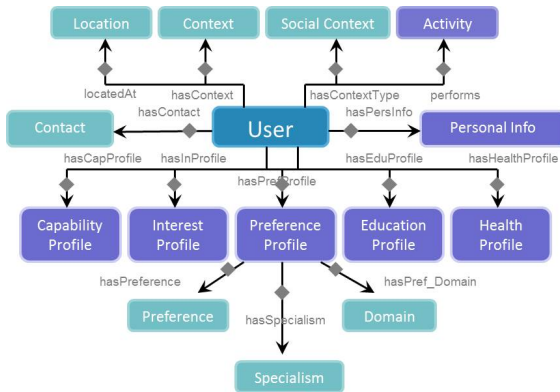


Fig. 2. Conceptual user profile models

To provide personalised services it is necessary to specify what services should be provided and how they should be delivered or presented for a specific type of user. Such knowledge is usually domain dependent and application specific. As a result, we have conducted knowledge acquisition in the context of the MobileSage HoD application scenario and extracted personalisation knowledge. Though there are different ways of representing personalization knowledge as discussed earlier, in MobileSage we have adopted Semantic Web Rule Language (SWRL) for personalisation knowledge representation (refer to Figure 4). This is in line with ontological user model representation, thus facilitating assimilation and combination of both semantic and rule based reasoning. Both user models and personalisation rules are stored in native OWL files.

Rules only establish cause-effect relationships. A reasoning engine is required to decide if the pre-conditions of a rule are met, thus leading to a consequence. In addition, the reasoning engine also needs to decide if the consequence of a previously fired rule results in another rule being fired. In this study we used an existing open-source semantic reasoning engine called Pellet 9 as the personalisation engine.

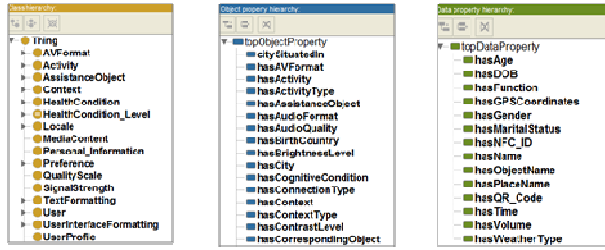


Fig. 3. Computational ontological user models

```

UserProfile(?up, hasHealthCondition(?up, Blind), hasPrimaryLanguage(?up, ?lang) -> HelpDelivery(PlayAudio),
hasMediaLanguage(PlayAudio, ?lang), hasMediaType(PlayAudio, Audio), hasMediaVolumeLevel(PlayAudio, VolLevel_5)
UserProfile(?up, hasPrimaryLanguage(?up, None) -> hasDefaultMediaLanguage(?up, English)
UserProfile(?up, hasPrimaryLanguage(?up, ?lang) -> MediaContent(Imagery), hasMediaLanguage(Imagery, ?lang)
UserProfile(?up, hasPreferredMediaContentType(?up, Text) -> HelpDelivery(DisplayText), hasMediaType(DisplayText, Text)
UserProfile(?up, hasHealthCondition(?up, Deaf), hasPreferredTextColour(?up, ?c) -> HelpDelivery(DisplayText),
hasMediaType(DisplayText, Text), hasTextColour(DisplayText, ?c), hasTextSize(DisplayText, TextSize_18)
  
```

Fig. 4. SWRL based personalisation rules for HoD

4.2 Usage Data Modelling and Adaptation Mechanism

Usage data are closely related to the service's functions and features of the applications concerned. In MobileSage we have characterised HoD services into four levels of interactions. Level One refers to all general user interactions such as *device events* and *user interface (UI) events*. *Device events* encompass any user interaction with the hardware of the Smartphone, such as Volume, Back and Menu button presses whilst *UI Events* are events in which the user has interacted with the UI of the HoD. Level Two refers to service interactions, e.g. what services the user has accessed and also what information or user input have been provided. For example, if the user enters the Search service and searches for restaurants, the Usage Log would record this as a *serviceEvent*, including what service was accessed, in this instance Search, and what the user's input was, i.e. "restaurants". Level Three refers to user profile interactions, i.e. which user profile properties have been accessed or changed, and the new values. For example, the user modifies the screen brightness, this would be logged as a *profileEvent* with the identifier "System_Profile_Brightness" and the new value, i.e. brightness level. The final level of event capture involves communication with the Content Management System (CMS). Level Four collects data in the context of "Server Response". When a service is accessed and a request for content is made to the CMS the request and a summary of the CMS response is recorded. These events summarise the service requesting content in addition to the user input. While the exact contents of the response can be recorded, a summary of the response is recorded too, which also specifies what modality is returned.

For each level of interaction a data model (refer to Figure 5) is developed to capture the usage data at the level. All usage data is stored locally on the device itself in an SQLite Database, which is later synchronised into a remote server (as described in Section 5.1). We have developed various data analysis methods to mine the usage data to extract user behaviour patterns. These include the general service usage, e.g. the

most used functions, the preferred interface configuration and location aware user interfaces (UI). Figure 6 displays the k-means clustering algorithm to determine the centroid of the locations that specific services were used based on the GPS information.

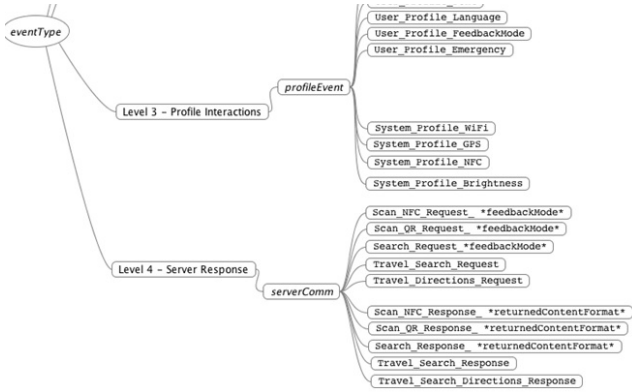


Fig. 5. An excerpt of event types and their correspondent models

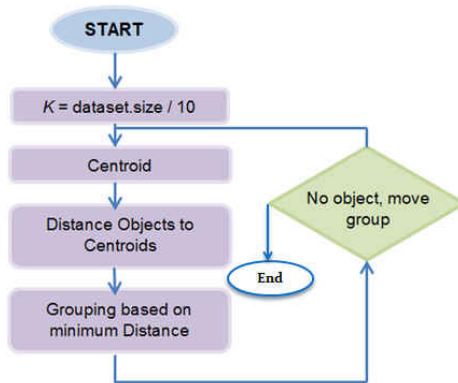


Fig. 6. The k-Means algorithm for centroid extraction

5 Implementation and Evaluation

We have partially implemented the personalisation services in the context of MobileSage project, as presented in Figure 7. The client side is the HoD services, which are implemented as a smartphone app deployed in an Android smartphone. HoD offers a number of services that are made available through an intuitive, adaptive UI 9. The server side includes four personalisation related services and also a *Content Management System* which provides content to the HoD application. All MobileSage assistive material is persisted at the CMS and can be provided in a range of different modalities including text, audio and video.

The four personalisation related services have been implemented in a REST-based (REpresentational State Transfer) software orientated architecture (SOA). The underlying technologies for User Profile Services and Personalisation Services include user ontologies, a set of pre-defined SWRL rules, and the employment of the Pellet reasoner. Semantic content such as instantiated user models and rules are managed through OWL-API². Usage data models and collected usage data are managed through SQLite on each client-side device while MySQL is used centrally on the server-side. Java web service development tools, specifically the JAX-RS - the Java API for RESTful web services and Jersey - an open-source, production quality framework, have been used to develop the four services. The development and testing environment included Eclipse Juno, MySQL Server 5.1 and Apache Tomcat 7.0.34. Messaging between services and client-side HoD have been represented as Java classes which are then mapped to JSON format by using the Google Gson library. The Android Developer Tools IDE supported development of the HoD Android application.

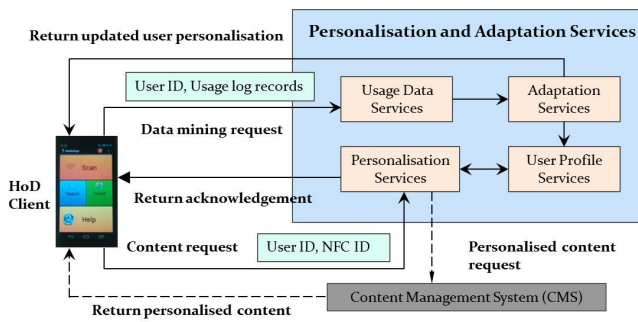


Fig. 7. Implementation of the personalisation services

5.1 Testing and Evaluation

The presented approach and developed underpinning technologies have been tested and evaluated in the Smart Environment Research lab at the University of Ulster in terms of assessing the efficiency and correctness of the HoD services and personalisation services. Three participants with different user profiles, as shown in Table 1, were instructed to use the HoD services to (1) seek help on purchasing a train ticket from an automated ticket machine and to (2) enable personalised route directions from a specified location to another place of interest. User 1 and 2 tested the first scenario while user 3 tested the second scenario. For the second scenario, over a period of twenty-four hours, a number of service activations were initiated through the HoD including search, scan, help and travel query terms. This activity was modelled to reflect a typical usage scenario of the HoD over a daily period.

² Open source OWL API <http://owlapi.sourceforge.net/>

Table 1. The three participants' user profiles

ID	Profile Name	Media Type Preference	Health Status	Sex	Default location	Age	Language Preference
1	Jane	Audio	Vision Impairment	Female	Northern Ireland	55	English
2	Jack	Text	Hearing Impairment	Male	UK	37	Spanish
3	Elizabeth	Video	None	Female	Northern Ireland	24	English

To test personalisation and adaptation the HoD usage data were collected and subsequently sent to data analysis services for pattern recognition, which in turn provided updated user models. Table 2 displays an example of the JSON representation of an updated user personalisation response.

To test performance, we recorded the formats of retrieved specific media and the time taken to retrieve information from the HoD service, as shown in Table 3. Initial results have shown that all returned media types match the ground truth of media preference of user profiles, and also the impact of the WiFi or 3G/GSM speed on service performance.

Table 2. Example user personalisation response

1	{ "searchClusters":["54.80232415814548,-5.683293549103988"],
2	"travelClusters":["54.827482000000025,-5.648274200000002","54.827482000
3	"nfcClusters":["54.8274820000000025,-5.648274200000002"],"qrClusters":[],
4	"addToFavourites":{"search":["Versailles"],"travel":["Giverny"]},
5	"serviceActivationsLast24":{"search":9,"scan":2,"help":0,"travel":0},
6	"serviceActivations":{"search":17,"scan":22,"help":16,"travel":12}}

Table 3. Testing and evaluation experiment results

ID	Media Preference	Returned Media Type	WiFi Speed ^a (kbps)	GSM Speed (kbps)	Time WiFi (ms)	Time GSM (ms)
1	Audio	Audio (4.7MB mp3)	DL 15267; UP 3861	DL 2488; UP 1433	6066	8384
2	Text	Text	DL 5035; UP 6563	DL 780; UP 1524	2245	3870
3	Video	Video (8.7MB 3gp)	DL 13737; UP 16171	DL 2319; UP 1512	6512	9532

6 Conclusions

This paper proposed an integrated approach and its associated service oriented system architecture for service personalisation of pervasive systems in smart environments. Relevant underpinning technologies such as behaviour analysis and personalisation have been developed based on semantic modelling, representation and reasoning. These services have been prototyped using the latest semantic technologies and service oriented software engineering, and further evaluated in well-designed use scenarios. Initial results have shown that the technologies and service oriented system are working. While further evidence will await large-scale evaluation in real world context with real users, current findings have proved the approach to be viable.

Acknowledgements. This work was undertaken as part of the EU AAL funded MobileSage project (ref: AAL-2011-3-50). The authors gratefully acknowledge the contributions from the members of the MobileSage consortium.

References

1. Anand, S., Mobasher, B.: Intelligent Techniques for Web Personalization. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–36. Springer, Heidelberg (2005)
2. Weld, D.S., Anderson, C., Domingos, P., Etzioni, O., Gajos, K., Lau, T., Wolfman, S., Automatically personalizing user interfaces. In: Proceedings of the 18th IJCAI Conference, pp.1613–1619 (2003)
3. Gallacher, S., Papadopoulou, E., Taylor, N., Williams, M.H.: Learning user preferences for adaptive pervasive environments: An incremental and temporal approach. ACM TAAS **8**(1), 5 (2013)
4. Chen, H., Finin, T., Joshi, A.: An Ontology for Context Aware Pervasive Computing Environments. The Knowledge Engineering Review **18**, 197–207 (2003)
5. Razmerita, L., Angehrn, A., Maedche, A., Ontology Based User Modeling for Knowledge Management Systems. In: Brusilovsky, P., Corbett, A., de Rosis, F. (eds.) UM 2003. LNCS (LNAI), vol. 2702, pp. 213–217. Springer, Heidelberg (2003)
6. Sutterer, M., Droegehorn, O., David, K., UPOS: User Profile Ontology with Situation-Dependent Preferences Support. In: Advances in Computer-Human Interaction, pp. 230–235 (2008)
7. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A Survey of Context Modelling and Reasoning Techniques. Pervasive and Mobile Computing **6**, 161–180 (2010)
8. Viviani, M., Bennani, N., Egyed-Zsigmond, E., A Survey on User Modeling in Multi-Application Environments. In: Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC), pp. 111–116 (2010)
9. Halbach, T., Schulz, T.: MobileSage - A Prototype Based Case Study Delivering Context-Aware, Personalized, On-Demand Help Content, in Advances in Human oriented and Personalized Mechanisms, Technologies, and Services, pp. 1–6 (2013)
10. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical Owl-DI Reasoner. Web Semantics: science, services and agents on the World Wide Web **5**, 51–53 (2007)

Extraction of Class Attributes from Online Encyclopedias

Hongzhi Guo¹(✉), Qingcai Chen², and Chunxiao Sun¹

¹ School of Computer Science and Technology, Xidian University,
Xi'an 710071, P.R. China

hzguo@xidian.edu.cn

² Shenzhen Graduate School, Harbin Institute of Technology,
Shenzhen 518055, P.R. China

qingcai.chen@gmail.com

Abstract. Class attributes are important resources in question answering, knowledge base building and semantic retrieval. In this paper, we propose an approach extracting class attributes from online encyclopedias. This approach combines the tolerance rough set model and semantic relatedness computing. Firstly, the implementation of the tolerance rough set model ensures a high precision of top- k extracted class attributes, and then the semantic relatedness computing improves the coverage of top- k extracted class attributes in order to achieve higher accuracy. Finally experiments on the extracted class attributes show the effectiveness of our approach.

Keywords: Class attribute extraction · Tolerance rough set · Semantic relatedness computing · Normalized google distance

1 Introduction

1.1 Motivations

Class attribute extraction is a branch of open information extraction and plays an important role in the fields of natural language processing (NLP) and information retrieval (IR) [1]. From a knowledge engineering perspective in artificial intelligence, class attributes are an interpretation of the target class. In the study of NLP, class attributes, as generalized semantic relations ($\langle \textit{Scientist}, \textit{Born} \rangle \Rightarrow (\textit{Person-BornIn-Place}, \textit{Person-BornOn-Date}, \dots)$) [2], which can be used to expand existing semantic relation sets, can be implemented in semantic relation extraction.

Furthermore, there are lots of queries and questions that request factual information in the study of IR and question answering (QA). Combined with their classes and entities, class attributes can be used not only in query expansion, but also to position the concrete results, and so that semantic retrieval can be realized. For example, there is a keyword-based query stream “*rose origin*” or a natural-language question “*what is the origin of rose ?*”, after some NLP

processing we find that “*origin*” is a prominent attribute of class “*flower*” to which the instance “*flower*” belongs. Then the origin information can be directly returned as an answer for the query or the question.

Recently, there has been lots of research on class attribute extraction emerged [3]. In source data selection, some of them take search engine query logs [4], and others take use of Web document collections. Besides, with the success of Wikipedia, information extraction from online encyclopedias becomes a new research focus [5]. Unfortunately most of the researchers focus on extracting from the infoboxes. The large scale instance contents are ignored. In extraction methods, existing work mostly adopts template matching methods based on statistical distributions, or supervised, semi-supervised learning methods. The accuracy of extracted attributes is ensured. Nevertheless, the sets of extracted class attributes have a low coverage, some potential important attributes are filtered out. To address these issues, a novel class attribute extraction approach using rough set and semantic relatedness is proposed in this paper.

Algorithm 1. Class attribute extraction using semantic relatedness.

Input: The document collection of instances, $D = d_1, d_2, \dots, d_m$; θ - the co-occurrence threshold; η - the NGD threshold.

Output: A class attribute collection, S .

- 1: Extract catalog labels from textual instance documents, and establish the raw candidate attribute set.
 - 2: Carry out preprocessing and word segmentation on the raw candidate attribute set, $\mathbb{A} = a_1, a_2, \dots, a_n$, build up the document-candidate attribute matrix (a $m \times n$ matrix).
 - 3: Compute the co-occurrences between candidate attribute pairs and construct attribute co-occurrence matrix (a $n \times n$ matrix).
 - 4: Compute the NGD values of candidate attribute pairs referring to the co-occurrence matrix.
 - 5: Given a co-occurrence threshold θ , generate a generalized approximation space for the attribute set, S' .
 - 6: Given the co-occurrence threshold θ and NGD threshold η , acquire attribute pairs satisfying the threshold from the co-occurrence matrix.
 - 7: Carry out set intersection on the results of steps 5 and 6, and return the final class attribute collection, S .
-

1.2 Contributions and Outlines

In this paper, we present a novel class attribute extraction approach. The class attribute extraction problem is abstracted as the determination of the generalized approximation space in Rough Set. Moreover, semantic relatedness computing is introduced to determine the boundary of class attribute set. Combined tolerance rough set with semantic relatedness, potential important semantic attributes can be mined from the candidate attribute set using our proposed method.

The rest of this paper is organized as follows: In Sec. 2, class attribute extraction methods unifying tolerance rough set and semantic relatedness are detailed; Sec. 3 gives a comprehensive evaluation of the proposed approach in this paper; After that, conclusions and discussions are described in Sec. 4.

2 Class Attribute Extraction Method

A class is a placeholder for a set of instances that share similar attributes [1]. Each entity has its own candidate attribute set, and those sets from the same class constitute a candidate class attribute set. The primary class attribute set, which serves as a primary description for the target class, is a subset of the candidate attribute set.

2.1 Tolerance Rough Set Model

In class attribute extraction, the attributes set \mathbb{A} of class C is denoted as the universe U , $U = \{a_1, a_2, \dots, a_n\} = \mathbb{A}$, and $D = \{d_1, d_2, \dots, d_m\}$ is the set of entity instance documents. The tolerance rough set model is implemented to model the class attribute extraction problem [6, 7], and the key issue is to define the tolerance class of class attributes, and the co-occurrences between different class attributes is introduced below.

Definition 1. Let $\mathfrak{R} = (U, I_\theta, v, P)$ be an approximation space over the universe \mathbb{A} , and $f_D(a_i, a_j)$ be the number of entries from D in which the attributes a_i, a_j occurs together. The uncertainty function I regarding the co-occurrence threshold θ is defined as

$$I_\theta(a_i) = \{a_j | f_D(a_i, a_j) \geq \theta\} \cup \{a_i\} \tag{1}$$

The function I_θ satisfies the conditions of being reflexive ($a_i \in I_\theta(a_i)$, for any $a_i \in \mathbb{A}$) and symmetric ($a_j \in I_\theta(a_i) \Leftrightarrow a_i \in I_\theta(a_j)$, for any $a_i, a_j \in \mathbb{A}$). And so the tolerance relation R can be defined by means of the uncertainty function ($a_i R a_j \Leftrightarrow a_j \in I_\theta(a_i)$, $R \subseteq \mathbb{A} \times \mathbb{A}$), $I_\theta(a_i)$ is the tolerance class of class attribute a_i . By varying the threshold θ , the degree of relatedness among the class attributes can be controlled. The vague inclusion v is defined as

$$v(X, Y) = \frac{|X \cap Y|}{|X|} \tag{2}$$

Definition 2. All tolerance classes of class attributes are structural subsets, $P(I_\theta(a_i)) = 1$ for any $a_i \in \mathbb{A}$. Finally the lower and upper approximations of any set $X \subseteq \mathbb{A}$ can be determined as (with regard to the tolerance space $\mathfrak{R} =$

$$(U, I_\theta, v, P): \quad \begin{aligned} L(\mathfrak{R}, X) &= \{a_i \in \mathbb{A} : v(I_\theta(a_i), X) = 1\}, \\ U(\mathfrak{R}, X) &= \{a_i \in \mathbb{A} : v(I_\theta(a_i), X) > 0\}. \end{aligned} \tag{3}$$

If we treat X as a category described vaguely by the class attributes it contains, $U(\mathfrak{R}, X)$ is the set of class attributes that share higher co-occurrence distributions with X .

2.2 Semantic Relatedness

Research and applications on semantic relatedness have been lasted for many years, and there have been a variety of computing methods raised so far, including Pointwise Mutual Information (PMI), Explicit Semantic Analysis [8], Normalized Google Distance (NGD) [9], etc. NGD is implemented here considering its effectiveness and popularity in information processing fields. A brief description on NGD is given below.

Given a keyword set, NGD takes use of the number of keyword hits returned by Google search engine to measure the semantic relatedness between them. In the sense of natural language, the keywords with the same or similar meanings often have a short NGD, and vice versa. For two terms t_1, t_2 , their NGD is defined as

$$NGD(t_1, t_2) = \frac{\max\{\lg f(t_1), \lg f(t_2)\} - \lg f(t_1, t_2)}{\lg M - \min\{\lg f(t_1), \lg f(t_2)\}}$$

Where M is the total

number of hits returned by Google search engine, $f(t_1)$ and $f(t_2)$ are the number of hits of items “ t_1 ” and “ t_2 ” respectively, $f(t_1, t_2)$ denotes the number of hits of item combination “ $t_1 t_2$ ”. If the items “ t_1 ” and “ t_2 ” never occur in the same web page, the NGD between them is infinite, and vice versa, if they always occur in the same web page, the NGD between them equals to zero.

Recently researchers have proved by experiments that semantic relatedness computing using NGD also applies to medium-sized corpus [10]. In class attribute extraction, online encyclopedia knowledge bases (Baidu in this paper) are chosen as the source of data. For a target class, the collection of instance documents is considered as a sample of online encyclopedia knowledge bases. The number of the instances of the target class is set as the total number of pages M , and the number of candidate attributes occurred in the instances of the target class is set as the number of hits in NGD, $f(t)$, the number of co-occurrences between candidate attributes is set as $f(t_1, t_2)$. In computing semantic relatedness between candidate attributes using NGD, its value usually varies between 0 and 1, the smaller the NGD value the higher semantic relatedness between candidate attributes, and vice versa.

2.3 Class Attribute Extraction Using TRS and Semantic Relatedness

The TRS model of class attribute extraction is mainly developed by a generalized approximation space, which is defined by the co-occurrences between candidate attributes. Semantically, the co-occurrence between two candidate attributes reflects the semantic relatedness between them. Ideally, the higher value of the co-occurrence between two attributes, the higher semantic relatedness between them, and vice versa. Nevertheless, during the experiment we find that there are still shortcomings only relying on the co-occurrence threshold. Further analysis shows that, this is because one of the two attributes almost appears in every instance

document, and the other hardly turns up without the former although it appears rarely (above the co-occurrence threshold). Semantic relatedness computing is introduced with co-occurrence threshold to complement each other, hoping that the candidate attributes which have the potential high semantic relatedness can be mined. Alg. 1 describes our class attribute extraction algorithm.

3 Evaluation and Discussion

3.1 Experiment Setting

Data: In the evaluation procedure, the extracted class attributes should be compared to some ground truth. Unfortunately, there still lacks of computer-processable ground truth for class attribute extraction until now, and we have to rely on manual evaluation. In our experiments, a random sample of around 72,537 textual instance pages from Baidu Baike is chosen as the data source. After some preprocessing, 6,472 instance documents constitute our final corpus. *Target classes:* Considering of the time intensive nature of manual accuracy judgements in information extraction [1], 10 classes are randomly chosen as the target classes. Table 1 shows the data distribution. In these 10 classes, each of them is specified to represent a part of the instances in the 10 ancestor classes in Baidu Baike, and the instance number varies from 444 (class “Prose”) to 732 (class “Basketball”), with a median of about 647 instances per class.

Table 1. Target Classes with set sizes and instance examples

Class	Size	Examples of Instances
Scientist	712	Weichang Qian, Albert Einstein, Xuesen Qian, Nikola Tesla
Prose	444	Ode to Cibi, Yueyang Tower, Moonlight over the Lotus Pond, To The Oak
The Three Kingdoms	701	Liang Zhuge, Cao Cao, Yun Zhao, Yu Guan
Oil Painting	665	Lonely, The Last Supper, The Boat Trackers of Volga River
Cooking	728	Braised pork in brown sauce, Sweet and Sour Spare Ribs, Buddha Jumps over the Wall
Organization	614	Mafia, VISA, Oracle Inc., United Nations, State-Owned Enterprises
Basketball	732	Kobe Bryant, SLAM DUNK, Yao Ming, Michael Jordan, NBA
Flower	711	Lily, Flower, Color Talks, Cymbidium, Rose
Definition	483	QQ Skins, Prime Number, Matrix, Chinese, Mouthrinse
Stock	682	Stock, Futures, Price to Earning Ratio, Warren E. Buffett, Huaxi Village

To verify the efficiency of the class attribute extraction algorithm in this paper, three criterions are implemented, recall, precision and coverage.

Recall: In the evaluation of class attribute extraction, a complete set of attributes for the target class is required in computing the recalling rate. Unfortunately, this complete set is often unavailable in an information extraction task. In comparison, we are more interested in the accuracy of the extracted class attributes in this paper.

Table 2. Correctness labels in precision evaluation

Label	Value	Example ([attribute, class])
vital	1.0	[life, scientist], [characteristic, flower]
okay	0.5	[legend, flower], [season, basketball]
wrong	0	[album, basketball], [related, scientist]

Precision: In evaluating the accuracy rate, every attribute is assigned to a correctness label during the assessments, which is similar to the methodology previously proposed by Eleen M. Voorhees in question answering evaluation [11]. Table 2 shows the correctness labels for manual assessments. To simplify the computation of the precision scores, the correctness labels are converted to numeric values. Therefore the precision of chosen N attributes ($precision@N$) is measured as the sum of the correctness values of the m attributes divided by N .

$$precision@N = \frac{\sum_{i=1}^N V_i}{N} \quad (4)$$

Coverage: Coverage is mainly used to evaluate whether our algorithm can mine the attributes which have latent semantic relatedness from the candidate attribute set of the target class. Different from the calculation of precision, coverage is more focused on the number of attributes associated with the target class in the returned result set (class attribute set without duplicates), the formula for calculating the coverage of chosen class attributes is:

$$C_{\mathbb{A}} = \frac{|\mathbb{A}_{Rel} \cap \mathbb{A}_{Res}|}{|\mathbb{A}_{Res}|} \quad (5)$$

where \mathbb{A} is the attribute set for the target class, $\mathbb{A}_{Rel} \subseteq \mathbb{A}$ is the attribute set associated with the target class, \mathbb{A}_{Res} denotes the returned class attribute set, and $|\cdot|$ denotes the number of the attribute set. Different from the assessments in precision evaluation, the assessors here only need two types of labels. An attribute is *relevant* if it is related to the description of the target class; *irrelevant* if it has no relationship with the target class. Similar to the precision evaluation, the correctness labels are converted to numeric values as shown in Table 3.

Table 3. Labeling rules in coverage evaluation

Label	Value	Example ([attribute, class])
relevant	1.0	[life, scientist], [season, basketball]
irrelevant	0	[album, basketball], [origin, scientist]

Parameter selection: In our class attribute extraction approach there are two parameters to be determined, co-occurrence threshold θ and NGD threshold η . The co-occurrence threshold is mainly used to define the tolerance class between candidate class attributes, and then determine the upper approximation of the

Table 4. Comparisons on Coverage and Precision between the attribute results using rule-based and statistical methods and our method in this paper

Class	Coverage		Precision	
	C_{RulSta}	C_{Sem}	P_{RulSta}	P_{Sem}
Scientist	0.8	0.95	0.625	0.6
Prose	0.7	0.9	0.5	0.575
Three Kingdoms	0.8	0.95	0.6	0.625
Oil Painting	0.85	0.9	0.525	0.475
Cooking	0.85	0.95	0.5	0.75
Organization	0.95	0.8	0.675	0.575
Basketball	0.85	0.8	0.65	0.625
Flower	0.9	0.85	0.625	0.725
Definition	0.75	0.85	0.55	0.625
Stock	0.85	0.8	0.675	0.65
Average-Class	0.83	0.875	0.5925	0.6225

class attribute set. With the change of the co-occurrence threshold, the final class attribute set also changes. The higher the co-occurrence threshold θ is, the fewer the final class attributes are, and the smaller the attribute collection space is, and also less attributes with potential semantic relatedness can be mined; and vice versa. However, more noise attributes appear along with a larger class attribute collection.

The introduction of NGD values between candidate class attributes is used to mine the attributes with potential semantic relatedness. The smaller the NGD threshold is, the higher the potential semantic relatedness between the mined candidate attributes is, and the less the number of final attributes is; and vice versa.

Table 5. NGD values with co-occurrence threshold $\theta = 4$ and a more than 20 attributes scale

Class	cooking	flower	stock	prose	organization
NGD	0.21	0.38	0.33	0.41	0.48

In this paper, we first evaluate the TRS method only using co-occurrence threshold. Regretfully the results are not always satisfied. In the result collection, there are some attribute pairs with high co-occurrences, but the NGD between them are extremely large. After the introduction of the NGD threshold, this situation can be significantly improved. Due to the formula for computing NGD, the smaller the differences between $\max\{\lg f(t_1), \lg f(t_2)\}$ and $\lg f(t_1, t_2)$ are, the smaller the calculated NGD value is. And this is very suitable for finding the candidate attributes which have relatively close occurrences. The co-occurrence threshold θ is set to 3, 4, 5, 7, 8, and the corresponding NGD values are computed respectively. The experiments show that with the increasing of θ , the numbers of the candidate attributes extracted using NGD decrease significantly. During this procedure, the NGD threshold η is selected. Table 5 lists the NGD values

of classes “cooking”, “prose”, “flower”, “stock” and “organization” with the co-occurrence threshold $\theta = 4$ and a more than 20 attribute scale. Finally, with a guarantee of the number and the quality of the mined attributes, the co-occurrence threshold θ is set to 4 and the corresponding NGD threshold is set to 0.48.

3.2 Comparison to Previous Results

The method in this paper brings in semantic relatedness computing to find these candidate attributes with high potential semantic relatedness. Therefore, this paper carry out evaluation on the coverage and precision of the method. Since there is little research on extracting class attributes using the instances of online encyclopedia knowledge bases, there is no uniform evaluation benchmark. To validate the effectiveness of the proposed approach in this paper, we have to compare our results with those using rule-based and statistical methods [7]. Based on the experimental setting discussed in Sec. 3.1, the experiments are conducted on 10 target classes using both methods, and finally top 20 extracted class attributes from both methods are compared on result coverage and precision separately. Table 4 illustrate the detailed comparison results.

Table 4 shows that the coverage of 6 classes (classes “scientist”, “The Three Kingdoms”, “cooking”, “definition”, “prose” and “oil painting”) of the 10 target classes has significantly improved while adopting the semantic relatedness. The coverage of the other 4 classes doesn’t improve, and some of them even descends. Against this issue further evaluation is conducted, and the results are illustrated in Fig. 1.

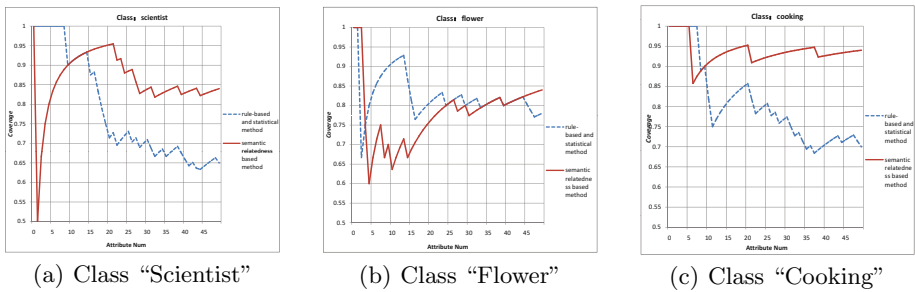


Fig. 1. Coverage comparisons of top 50 results between using rule-based and statistical method and our method

Due to the comparisons in Fig. 1, with the increasing of extracted result number(1-50), the coverage of our approach changes a little and still maintains a high level, however the coverage using ruled-based and statistical method decreases significantly (Fig. 1(a) and Fig. 1(c)) or shows an descending trend. Further experiments on other class corpus show a similar result. All these experiments prove that the coverage of top 20 results of some classes does not increase,

this is mainly due to the number of returned results. And the experimental results of the average coverage of all target classes also demonstrate the advantage of our method in the result coverage, and this advantage appears more obvious with the increasing of the returned results.

Moreover, compared with those using the rule-based and statistical methods the precision in this paper doesn't decrease. In the comparisons of the top 20 results, the precision of some classes doesn't decrease and even rises up, and the average precision of all classes also rises. The reason is that, after the coverage of extracted attributes rises up by adopting the semantic relatedness, the precision will increase accordingly with more class attributes being covered. Of course, since the extracted candidate class attributes are not disposed with synonyms merging and similarity computing, there is still a gap between them and those using rule-based and statistical methods.

Furthermore, in the coverage evaluation of top 5 extracted class attributes as shown in Fig. 1, there have been a sudden drop in coverage in classes "scientist" and "flower". The analysis of the annotation results shows that there are some unrelated attributes existed in top 5 extracted attributes of both classes, and they pull down the top 5 coverage.

4 Conclusion

In this paper, we have presented an approach extracting class attributes from online encyclopedias. Our approach firstly uses the tolerance rough set model to represent the class attribute extraction problem, and candidate class attribute sets are extracted based on this model; after that, semantic relatedness computing using NGD is implemented on the candidate class attribute sets. The experimental results show that the implementation of the tolerance rough set model ensures a high precision of top- k extracted class attributes. The semantic relatedness computing improves the coverage of top- k extracted class attributes in order to achieve a higher accuracy. Though a good performance is reached in this paper, there are still lots of work need to do, such as automatic attribute contents extraction, and semantic retrieval using class attributes and attribute contents.

Acknowledgments. This paper was supported by the Natural Science Foundation of ShaanXi (2013JQ8037), the Fundamental Research Fund for the Central Universities (K5051303002), and the science and technology plan of Shenzhen (JC201005260118A).

References

1. Pasca, M., Durme, B.V.: What you seek is what you get: extraction of class attributes from query logs. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence Hyderabad, India, pp. 2832–2837(2007)
2. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: The 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp. 697–706 (2007)

3. Pasca, M.: Open-Domain Fine-Grained Class Extraction from Web Search Queries. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 403–414 (2013)
4. Yoshinaga, N., Torisawa, K.: Open-Domain Attribute-Value Acquisition from Semi-Structured Texts. In: Proceedings of the Workshop on Ontolex, pp. 55–66 (2007)
5. Medelyan, O., Milne, D., Legg, C., Witten, I.H.: Mining meaning from Wikipedia. *International Journal of Human-Computer Studies* **67**, 716–754 (2009)
6. Ngo, C.L., Nguyen, H.S.: A Tolerance Rough Set Approach to Clustering Web Search Results. In: The 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, pp. 515–517 (2004)
7. Guo, H.Z., Chen, Q.C., Cui, L., Wand, X.L.: Tolerance rough set based attribute extraction approach for multiple semantic knowledge base integration. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **19**(4), 659–684 (2011)
8. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 16060–1611 (2007)
9. Cilibrasi, R.L., Vitanyi, P.M.B.: The Google Similarity Distance. *IEEE Trans. on Knowl. and Data Eng.* **19**, 370–383 (2007)
10. Pedersen, T., Kulkarni, A.: Discovering Identities in Web Contexts with Unsupervised Clustering. In: Proceedings of the IJCAI 2007 Workshop on Analytics for Noisy Unstructured Text Data, Hyderabad, India, pp. 23–30 (2007)
11. Voorhees, E.M.: Evaluating Answers to Definition Questions. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 2, Edmonton, Canada, pp. 109–111 (2003)

Selective Ensemble of RBFNNs Based on Improved Negative Correlation Learning

Hongjie Xing^(✉), Lifei Liu, and Sen Li

Key Laboratory of Machine Learning and Computational Intelligence,
College of Mathematics and Computer Science, Hebei University,
Baoding 071002, Hebei Province, China
hjxing@hbu.edu.cn

Abstract. In this paper, a novel selective ensemble method based on the improved negative correlation learning is proposed. To make the proposed ensemble strategy more robust against noise, correntropy is utilized to substitute mean square error (MSE). Moreover, an L1-norm based regularization term of ensemble weights is incorporated into the objective function of the proposed ensemble strategy to fulfill the task of selective ensemble. The half-quadratic optimization technique and the surrogate function method are used to solve the optimization problem of the proposed ensemble strategy. Experimental results on two synthetic data sets and the five benchmark data sets demonstrate that the proposed method is superior to the single radial basis function neural network (RBFNN).

Keywords: Selective ensemble · RBFNN · Correntropy · Negative correlation learning

1 Introduction

Neural network ensemble is first proposed by Hansen and Salamon [1]. They demonstrated that the generalization performance of a neural network system can be significantly improved by combining a number of neural networks. Generally, there are two steps to construct a neural network ensemble. First, several component networks are trained. Second, a certain combining strategy is applied to ensemble the outputs of the trained networks.

Recently, Liu and Yao [2,3] proposed negative correlation learning (NCL) and applied it to construct an efficient neural network ensemble. To make NCL faster, Chan and Kasabov [4] proposed a new NCL method, which is easy to implement, requires less communication, and is applicable to combining heterogeneous networks. Based on NCL, Islam et al. [5] proposed two cooperative ensemble learning algorithms, namely, NegBagg and NegBoost. The two algorithms can incrementally train different component networks using NCL. Chen and Yao [6] introduced an L2-norm based regularization term of connecting weights into NCL to enhance the anti-noise ability of NCL. Alhamdoosh and Wang [7] utilized NCL to construct the ensemble of random vector functional link networks to achieve a faster neural network ensemble.

Although NCL and its variants have achieved better performance, there are still two issues that need to be addressed. One is that NCL is sensitive to noise [6], which makes it prone to over-fitting. The other is that NCL has to combine all the provided component networks. As stated in literature [8], it is better to ensemble part of component networks rather than all of them.

To deal with the above-mentioned two disadvantages of NCL, an improved NCL based selective ensemble strategy is proposed. The main contributions of the proposed ensemble strategy are listed as follows:

- In contrast to NCL, correntropy is used to substitute mean square error (MSE) to enhance the anti-noise ability of NCL;
- An L1-norm based regularization term is introduced into the objective function of the proposed ensemble strategy to fulfill selective ensemble.

The paper is organized as follows. The related works, i.e., radial basis function neural network (RBFNN), NCL, and correntropy are briefly reviewed in Section 2. The proposed ensemble strategy is expatiated in Section 3. The results of the experiments are reported in Section 4. The conclusion is given in Section 5.

2 Preliminaries

In this section, RBFNN, NCL, and correntropy are briefly reviewed.

2.1 RBFNN

RBFNN is a three-layer feedforward neural network with one hidden layer [9]. The parametric model for a multiple-input and single-output RBFNN can be expressed as

$$\begin{aligned}
 f(\mathbf{x}) &= \sum_{k=1}^K w_k \phi_k(\mathbf{x}) + w_0 \\
 &= \sum_{k=1}^K w_k \exp \left\{ -\sum_{p=1}^d \frac{1}{2} \left(\frac{x_p - c_{kp}}{\sigma_k} \right)^2 \right\} + w_0,
 \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \dots, x_d)^T \in \mathfrak{R}^d$ is the input vector, $\mathbf{c}_k = (c_{k1}, c_{k2}, \dots, c_{kd})^T$ is the prototype vector of the k th hidden unit, while σ_k is the width parameter. Moreover, w_k is the weight connecting the k th unit of the hidden layer and the output node while constant w_0 is the bias term.

There are several different algorithms for training an RBFNN. The three most commonly used methods are backpropagation [10], two-step learning [11], and evolutionary learning [12]. In this paper, the two-step learning method is adopted. In the first step, the centers of the clusters, \mathbf{c}_k ($k = 1, 2, \dots, K$), obtained by the FCM clustering algorithm, are directly used as the values for their corresponding centers of the hidden units. Moreover, the values of the width parameter σ_k ($k = 1, 2, \dots, K$) are calculated by

$$\sigma_k = \zeta \min(\|\mathbf{c}_k - \mathbf{c}_l\|), \quad l = 1, 2, \dots, K \text{ and } l \neq k, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm, while ζ is a parameter deciding the degree of overlap between the clusters and which is assigned the value 0.85 in this paper. In the second step, the weights connecting the hidden layer and the output layer, together with the bias term, are determined by the linear least-squared method.

2.2 NCL

Negative correlation learning (NCL) is a neural network ensemble approach [2,3]. Its error function consists of two parts. The first part is used for measuring individual training error of each network, while the second part is utilized to evaluate the negative correlation of each network's error with errors for the rest ensemble. Given the training set $\{\mathbf{x}_n, y_n\}_{n=1}^N$, the ensemble error of NCL for the i th component network is given by

$$E_i = \sum_{n=1}^N e_i(\mathbf{x}_n) = \sum_{n=1}^N \left\{ \frac{1}{2} [f_i(\mathbf{x}_n) - y_n]^2 + \lambda p_i(\mathbf{x}_n) \right\}, \tag{3}$$

where $f_i(\mathbf{x}_n)$ denotes the output of the i th component network upon the n th sample \mathbf{x}_n , λ is a weighting parameter upon the penalty term $p_i(\mathbf{x}_n)$ with

$$\begin{aligned} p_i(\mathbf{x}_n) &= [f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n)] \sum_{j \neq i} [f_j(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n)] \\ &= -[f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n)]^2 \end{aligned} \tag{4}$$

$f_{ens}(\mathbf{x}_n)$ in (4) represents the output of the network ensemble on \mathbf{x}_n

$$f_{ens}(\mathbf{x}_n) = \frac{1}{M} \sum_{i=1}^M f_i(\mathbf{x}_n), \tag{5}$$

where M is the number of component networks in the ensemble.

Therefore, the error of the whole ensemble network is given by

$$\begin{aligned} E_{ens} &= \frac{1}{M} \sum_{i=1}^M E_i \\ &= \frac{1}{M} \sum_{i=1}^M \sum_{n=1}^N \left\{ \frac{[f_i(\mathbf{x}_n) - y_n]^2}{2} - \lambda [f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n)]^2 \right\}. \end{aligned} \tag{6}$$

2.3 Correntropy

Correntropy [13] is considered as a generalized correlation function [14]. It is a similarity measure by studying the interaction of the given feature vectors. Given two arbitrary random variables A and B , their correntropy can be defined as

$$V_{\delta}(A, B) = E[\kappa_{\delta}(A - B)], \tag{7}$$

where $\kappa_{\delta}(\cdot)$ is the kernel function that satisfies Mercer's theorem [15] and $E[\cdot]$ denotes the mathematical expectation.

In the paper, we only consider the Gaussian kernel with finite number of data samples $\{(a_n, b_n)\}_{n=1}^N$. Thus, correntropy can be estimated by

$$\hat{V}_{N,\delta}(A, B) = \frac{1}{N} \sum_{n=1}^N \kappa_{\delta}(a_n - b_n), \quad (8)$$

where $\kappa_{\delta}(\cdot)$ is given by

$$\kappa_{\delta}(a_n - b_n) = G(a_n - b_n) = \exp\left\{-\frac{(a_n - b_n)^2}{2\delta^2}\right\}. \quad (9)$$

Therefore, (8) can be rewritten as

$$\hat{V}_{N,\sigma}(A, B) = \frac{1}{N} \sum_{n=1}^N G(a_n - b_n), \quad (10)$$

The maximum of correntropy function (7) is called the maximum correntropy criterion (MCC) [14]. Since correntropy is insensitive to noise, it is superior to MSE when there are impulsive noises in training samples [14].

3 Selective Ensemble Based on the Improved Negative Correlation Learning

In this section, the improved negative correlation learning based selective ensemble for RBFNN is introduced. Moreover, its optimization method is presented.

To assign different weights for different component networks, the output of network ensemble in (5) is revised as

$$\tilde{f}_{ens}(\mathbf{x}_n) = \sum_{i=1}^M \alpha_i f_i(\mathbf{x}_n), \quad (11)$$

where the weights α_i ($i = 1, 2, \dots, M$) satisfy

$$\sum_{i=1}^M \alpha_i = 1, \quad \alpha_i \geq 0. \quad (12)$$

Furthermore, to make the proposed ensemble strategy more robust against noise, the minimization of the first term in (6) is replaced with maximizing the correntropy between outputs of component networks and target output. Moreover, to fulfill the selective ensemble, an L1-norm based regularization term is incorporated. The objective function of the improved NCL is expressed as

$$\max_{\mathbf{w}, \alpha} \frac{1}{M} \sum_{i=1}^M \sum_{n=1}^N G(f_i(\mathbf{x}_n) - y_n) - \frac{\lambda}{M} \sum_{i=1}^M \sum_{n=1}^N \tilde{p}_i(\mathbf{x}_n) - \gamma \|\alpha\|_1, \quad (13)$$

where the correntropy function is defined as

$$G(f_i(\mathbf{x}_n) - y_n) = \exp\left\{-\frac{[f_i(\mathbf{x}_n) - y_n]^2}{2\delta^2}\right\}, \quad (14)$$

$\tilde{p}_i(\mathbf{x}_n)$ is defined as

$$\begin{aligned} \tilde{p}_i(\mathbf{x}_n) &= \left[f_i(\mathbf{x}_n) - \tilde{f}_{ens}(\mathbf{x}_n) \right] \sum_{j \neq i} \left[f_j(\mathbf{x}_n) - \tilde{f}_{ens}(\mathbf{x}_n) \right] \\ &= - \left[f_i(\mathbf{x}_n) - \tilde{f}_{ens}(\mathbf{x}_n) \right]^2, \end{aligned} \tag{15}$$

γ is a regularization parameter, and $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_M)^T$. Throughout the paper, the parameter λ is taken as 1.

There are many approaches for solving the optimization problem (13) in the literature, for example, half-quadratic optimization technique [16,17], expectation-maximization (EM) method [18], and gradient-based method [13]. In this paper, the half-quadratic optimization technique is utilized.

According to the theory of convex conjugated functions [17,19], the following proposition [16] exists.

Proposition 1 For $G(\mathbf{z}) = \exp \left\{ -\frac{\|\mathbf{z}\|^2}{2\delta^2} \right\}$, there exists a convex conjugated function

φ , such that

$$G(\mathbf{z}) = \sup_{\alpha \in \mathfrak{R}} \left(\alpha \frac{\|\mathbf{z}\|^2}{2\delta^2} - \varphi(\alpha) \right). \tag{16}$$

Moreover, for a fixed \mathbf{z} , the supremum is reached at $\alpha = -G(\mathbf{z})$ [16].

Therefore, by introducing (16) into the objective function of (17), the following augmented objective function can be obtained

$$\max_{\mathbf{w}, \mathbf{a}, \mathbf{p}} \frac{1}{M} \sum_{i=1}^M \sum_{n=1}^N \left\{ p_m \frac{[f_i(\mathbf{x}_n) - y_n]^2}{2\delta^2} - \varphi(p_m) \right\} + \frac{1}{M} \sum_{i=1}^M \sum_{n=1}^N [f_i(\mathbf{x}_n) - \tilde{f}_{ens}(\mathbf{x}_n)]^2 - \gamma \|\mathbf{a}\| \tag{17}$$

where $\mathbf{P} = (p_{in})_{M \times N} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M)^T$ stores the auxiliary variables appeared in the half-quadratic optimization.

According to the half-quadratic optimization technique, the local optimal solution of (17) can be calculated iteratively by

$$p_{in}^{\tau+1} = -G \left(f_i^\tau(\mathbf{x}_n) - y_n \right), \tag{18}$$

$$\mathbf{w}^{\tau+1} = \arg \max_{\mathbf{w}} \frac{1}{M} \sum_{i,n} \left\{ p_{in}^\tau \frac{[f_i^\tau(\mathbf{x}_n) - y_n]^2}{2\delta^2} + [f_i^\tau(\mathbf{x}_n) - \tilde{f}_{ens}^\tau(\mathbf{x}_n)]^2 \right\}, \tag{19}$$

$$\mathbf{a}^{\tau+1} = \arg \max_{\mathbf{a}} \frac{1}{M} \sum_{i,n} [f_i^\tau(\mathbf{x}_n) - \tilde{f}_{ens}^\tau(\mathbf{x}_n)]^2 - \gamma \|\mathbf{a}\|, \tag{20}$$

where τ denotes the τ th iteration.

As for RBFNN, (19) can be rewritten as

$$\mathbf{w}_i^{\tau+1} = \arg \max_{\mathbf{w}_i} \frac{1}{M} \sum_i \left[\frac{1}{2\delta^2} (\Phi_i \mathbf{w}_i - \mathbf{y})^T \text{diag}(\mathbf{p}_i^{\tau+1}) (\Phi_i \mathbf{w}_i - \mathbf{y}) + \sum_{j,l} \alpha_j \alpha_l (\Phi_i \mathbf{w}_i - \Phi_j \mathbf{w}_j)^T (\Phi_i \mathbf{w}_i - \Phi_l \mathbf{w}_l) \right], \quad (21)$$

where $\mathbf{w}_i = (w_{i0}, w_{i1}, \dots, w_{iN_i})^T$ denotes the weight vector of the i th component network with its first element w_{i0} and N_i respectively representing the bias term and the number of hidden units, the output matrix of the hidden units are given by

$$\Phi_i = \begin{bmatrix} 1 & \phi_{i1}(\mathbf{x}_1) & \cdots & \phi_{i1}(\mathbf{x}_N) \\ 1 & \phi_{i2}(\mathbf{x}_1) & \cdots & \phi_{i2}(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_{iN_i}(\mathbf{x}_1) & \cdots & \phi_{iN_i}(\mathbf{x}_N) \end{bmatrix}, \quad (22)$$

and $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$.

Hence, the optimal solution of (21) can be obtained as follows

$$\mathbf{w}_i^{\tau+1} = \left[\Phi_i^T \left(\frac{1}{\delta^2} \text{diag}(\mathbf{p}_i^{\tau+1}) + 2 \sum_{j,l} \alpha_j \alpha_l \mathbf{I}_N \right) \Phi_i \right]^{-1} \left[\frac{1}{\delta^2} \Phi_i^T \text{diag}(\mathbf{p}_i^{\tau+1}) \mathbf{y} + \sum_{j,l} \alpha_j \alpha_l (\Phi_i^T \Phi_j \mathbf{w}_j^\tau + \Phi_i^T \Phi_l \mathbf{w}_l^\tau) \right], \quad (23)$$

where \mathbf{I}_N is an identity matrix of order N .

Moreover, the optimization problem (20) can be solved by the surrogate function method [20]. According to literature [20], the following inequality holds

$$-\|\mathbf{a}\|_1 \geq -\frac{1}{2} \left(\sum_{i=1}^M \frac{\alpha_i^2}{|\alpha_i^\tau|} + \sum_{i=1}^M |\alpha_i^\tau| \right), \quad (24)$$

Thus, the optimal solution of (20) can be obtained by solving the following surrogate optimization problem

$$\begin{aligned} \mathbf{a}^{\tau+1} &= \arg \max_{\mathbf{a}} \left[\frac{1}{M} \sum_n \|\mathbf{f}_n^\tau - \mathbf{1}_M (\mathbf{f}_n^\tau)^T \mathbf{a}\|_2^2 - \frac{\gamma}{2} \left(\sum_i \frac{\alpha_i^2}{|\alpha_i^\tau|} + \sum_i |\alpha_i^\tau| \right) \right], \\ &= \arg \max_{\mathbf{a}} \left[\frac{1}{M} \sum_n \|\mathbf{f}_n^\tau - \mathbf{1}_M (\mathbf{f}_n^\tau)^T \mathbf{a}\|_2^2 - \frac{\gamma}{2} \sum_i \frac{\alpha_i^2}{|\alpha_i^\tau|} \right], \\ &= \arg \max_{\mathbf{a}} \left[\frac{1}{M} \sum_n \|\mathbf{f}_n^\tau - \mathbf{1}_M (\mathbf{f}_n^\tau)^T \mathbf{a}\|_2^2 - \frac{\gamma}{2} \mathbf{a}^T \mathbf{U} \mathbf{a} \right], \end{aligned} \quad (25)$$

where $\mathbf{f}_n^\tau = (f_1^\tau(\mathbf{x}_n), f_2^\tau(\mathbf{x}_n), \dots, f_M^\tau(\mathbf{x}_n))^T$, $\mathbf{1}_M$ is a column vector with its elements are all one, and

$$\mathbf{U} = \text{diag}\left(|\alpha_1^\tau|^{-1}, |\alpha_2^\tau|^{-1}, \dots, |\alpha_M^\tau|^{-1}\right). \quad (26)$$

Therefore, we can get

$$\boldsymbol{\alpha}^{\tau+1} = \left[\sum_n \mathbf{f}_n^\tau \mathbf{1}_M^T \mathbf{1}_M (\mathbf{f}_n^\tau)^T - \frac{M}{2} \gamma \mathbf{U} \right]^{-1} \left(\sum_n \mathbf{f}_n^\tau \mathbf{1}_M^T \mathbf{f}_n^\tau \right). \quad (27)$$

4 Experimental Results

In this section, the performance of the proposed method, i.e., SERBFNN is compared with that of the single RBFNN on two synthetic data sets and five benchmark data sets. The error functions are all root mean square error (RMSE). The input vectors of a given data set are scaled to mean zero and unit variance, while the output vectors are normalized to $[0,1]$.

For SERBFNN, the coefficient of the regularization term γ is taken as 1 in the following experiments. Moreover, the width parameter δ in the correntropy function is determined by the Silverman's rule [21]

$$\delta = 1.06 \times \min\{\sigma_E, R/1.34\} \times N^{-1/5}, \quad (28)$$

where σ_E and R are the standard deviation and the error interquartile range of the error between the ensemble output and the target output, respectively. In addition, the maximum number of iterations for SERBFNN is set to be 50.

4.1 Synthetic Data Sets

In this subsection, two synthetic regression data sets are utilized to validate the proposed method. The description of them is presented as follows.

Sinc: This synthetic data set is generated by the function $y = \text{sinc}(x) = \frac{\sin(x)}{x} + \rho$,

where ρ is a Gaussian distributed noise. For each noise level, we generate a set of data points $\{(x_i, y_i)\}_{i=1}^{100}$ with x_i drawn uniformly from $[-10,10]$.

Func: This artificial data set is generated by the function $y(x_1, x_2) = x_1 \exp\{-(x_1^2 + x_2^2)\} + \rho$, where ρ is also a Gaussian distributed noise. For the certain noise level $\rho \sim N(0,0.16)$, 200 data points are constructed by randomly chosen from the evenly spaced 30×30 on $[-2,2]$.

The settings of parameters for the two methods upon the two data sets are summarized in Table 1. For SERBFNN, the number of hidden units in each component networks is randomly selected from $[5,25]$. Moreover, the number of the remaining component networks of SERBFNN is also included in Table 1.

Table 1. Settings of parameters for the two methods upon the two synthetic data sets

Datasets	RBFNN		SERBFNN	
	N_H	M	ϵ	N_S
<i>Sinc</i>	20	50	0.01	38
<i>Func</i>	25	20	0.01	18

Note: N_H -Number of hidden units; M -Number of component networks; ϵ -Threshold for deleting redundant component networks; N_S -Number of remaining component networks.

Figure 1 demonstrates the results of the two methods upon *Sinc* with two different noise levels. In Figure 1(a), the regression errors of RBFNN and SERBFNN are 0.1133 and 0.0771. Moreover, in Figure 1(b), the errors for RBFNN and SERBFNN are 0.1602 and 0.0986. Therefore, SERBFNN is more robust against noise on *Sinc* than RBFNN.

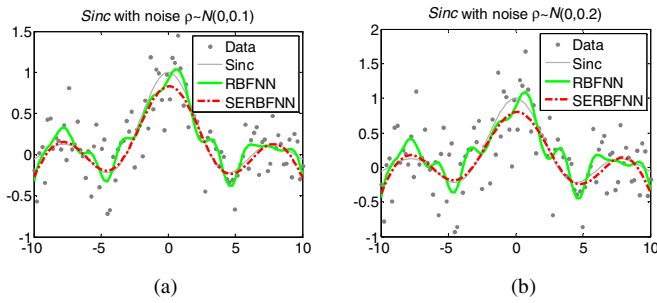


Fig. 1. The regression results of the two methods on *Sinc*. (a) *Sinc* with Gaussian noise $\rho \sim N(0, 0.1)$; (b) *Sinc* with Gaussian noise $\rho \sim N(0, 0.2)$.

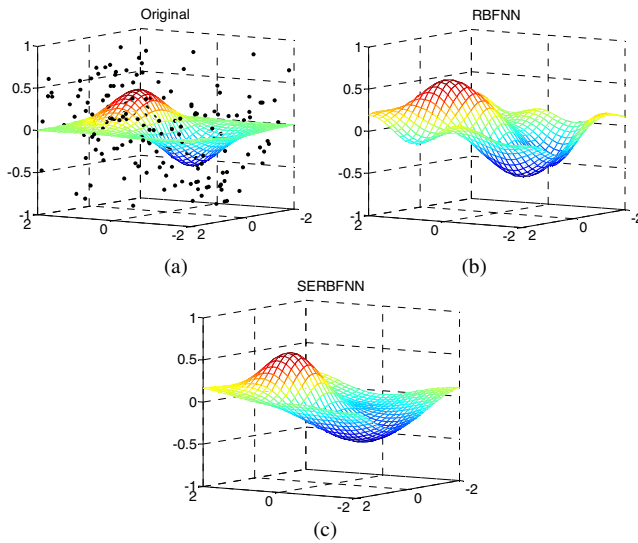


Fig. 2. The regression results of the two methods on *Func* with Gaussian noise $\rho \sim N(0, 0.16)$. (a) The original function and the polluted training samples; (b) The result of RBFNN; (c) The result of SERBFNN.

Figure 2 demonstrates the results of the two approaches upon $Func$. The regression errors of RBFNN and SERBFNN are 0.4158, and 0.3437. Then, we can find that SERBFNN is more robust against noise upon $Func$ than RBFNN.

4.2 Benchmark Data Sets

In the following comparisons, the parameter of RBFNN, i.e., number of hidden units N_H , is chosen by the 5-fold cross-validation on training set. The optimal number of hidden units for RBFNN is chosen from $\{5, 10, \dots, 50\}$. For SERBFNN, the number of hidden units in each component networks is randomly selected from $[5, 50]$. All the data sets are chosen from the UCI repository of machine learning databases [22]. For each data set, 50% samples are randomly chosen for training while the rest 50% are used for testing.

For the two methods, 20 trials are conducted on each data set and their corresponding average results are reported. The average training and testing errors together with their corresponding standard deviations are reported in Table 2.

Table 2. The results of the two methods on the five benchmark regression data sets

Datasets	RBFNN		SERBFNN	
	E_{train}	E_{test}	E_{train}	E_{test}
<i>Housing</i>	3.68 ± 0.11	5.34 ± 0.17	3.38 ± 0.07	5.24 ± 0.23
<i>Ozone</i>	4.32 ± 0.29	4.20 ± 0.15	3.92 ± 0.07	4.08 ± 0.12
<i>Santafe1</i>	12.06 ± 1.03	18.98 ± 1.48	10.94 ± 0.39	17.98 ± 1.73
<i>Servo</i>	0.53 ± 0.01	1.10 ± 0.01	0.66 ± 0.05	1.00 ± 0.02
<i>Wine Red</i>	0.68 ± 0.26	0.67 ± 0.00	0.62 ± 0.14	0.59 ± 0.00

Note: E_{train} -Training RMSE; E_{test} -Testing RMSE

It is shown in Table 2 that the proposed SERBFNN outperforms RBFNN on all the five data sets. Moreover, taking average testing errors into consideration, the values of standard deviation in Table 2 show that SERBFNN is more stable than RBFNN.

5 Conclusions

To make negative correlation learning (NCL) more robust against noise, a correntropy based objective function is utilized to replace mean square error (MSE) in NCL. Moreover, an L1-norm based regularization term of combination weights of the ensemble is added into the objective function of NCL to fulfill the selective ensemble. Half-quadratic optimization technique and surrogate function method are used to solve the optimization problem of the proposed ensemble strategy. Finally, the effectiveness of the presented selective ensemble is validated on two synthetic data sets and five benchmark data sets.

Acknowledgements. This work is supported by National Nature Science Foundation of China (No. 60903089, 61473111), Natural Science Foundation of Hebei Province (No. F2013201060, 2012201023).

References

1. Hansen, L.K., Salamon, P.S.: Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(10), 993–1001 (1990)
2. Liu, Y., Yao, X.: Ensemble learning via negative correlation. *Neural Networks* **12**(10), 1399–1404 (1999)
3. Liu, Y., Yao, X.: Simultaneous training of negatively correlated neural networks in ensemble. *IEEE Transactions on Systems, Man, and Cybernetics-Part B* **29**(6), 716–725 (1999)
4. Chan, Z.S., Kasavov, N.: Fast neural network ensemble learning via negative-correlation data correction. *IEEE Transactions on Neural Networks* **16**(6), 1707–1710 (2005)
5. Islam, M.M., Yao, X., Nirjon, S.M.S., Islam, M.A., Murase, K.: Bagging and Boosting negatively correlated neural networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* **38**(3), 771–784 (2008)
6. Chen, H., Yao, X.: Regularized negative correlation learning for neural network ensembles. *IEEE Transactions on Neural Networks* **20**(12), 1962–1979 (2009)
7. Alhamdoosh, M., Wang, D.: Fast decorrelated neural network ensembles with random weights. *Information Sciences* **264**, 104–117 (2014)
8. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: many could be better than all. *Artificial Intelligence* **137**(1-2), 239–263 (2002)
9. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice-Hall, New York (1999)
10. Rumelhart, D.E., McClelland, J.L.: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. MIT Press, Cambridge (1986)
11. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford (1995)
12. Chen, S., Wu, Y., Luk, B.L.: Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks* **10**(5), 1239–1243 (1999)
13. Liu, W., Pokharel, P.P., Principe, J.C.: Correntropy: properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Process* **55**(11), 5286–5297 (2007)
14. Santamaria, I., Pokharel, P.P., Principe, J.C.: Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Transactions on Signal Process* **54**(6), 2187–2197 (2006)
15. Vapnik, V.: *The nature of statistical learning theory*. Springer, New York (1995)
16. Yuan, X, Hu, B.G.: Robust feature extraction via information theoretic learning. In: *Proceedings of The 26th Annual International Conference on Machine Learning*, pp. 1193–1200. ACM, New York (2009)
17. Rockfellar, R.: *Convex analysis*. Princeton University Press, Princeton (1970)
18. Yang, S.-H., Zha, H., Zhou, S., Hu, B.-G.: Variational Graph Embedding for Globally and Locally Consistent Feature Extraction. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) *ECML PKDD 2009, Part II. LNCS*, vol. 5782, pp. 538–553. Springer, Heidelberg (2009)

19. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
20. Lannge, K., Hunter, D., Yang, I.: Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics* **9**(1), 1–59 (2000)
21. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
22. Blake C.L., Merz C.J., UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA (1998). <http://www.ics.uci.edu/mllearn/MLRepository.html>

A Two-Phase RBF-ELM Learning Algorithm

Junhai Zhai¹(✉), Wenxiang Hu¹, and Sufang Zhang²

¹ College of Mathematics and Computer Science,
Hebei University, Baoding 071002, China
mczjh@126.com

² Teaching and Research of Section of Mathematics,
Hebei Information Engineering School, Baoding 071000, China

Abstract. A variant of extreme learning machine (ELM) named RBF-ELM was proposed by Huang et al. in 2004. The RBF-ELM is tailored for radial basis function (RBF) networks. Similar to ELM, RBF-ELM also employs randomized method to initialize the centers and widths of RBF kernels, and analytically calculate the output weights of RBF networks. In this paper, we proposed a two-phase RBF-ELM learning algorithm, which only randomly initializes the width parameters. The center parameters are determined by an instance selection method. The first phase of the proposed algorithm is to select the centers of the RBF network rather than randomly initializing. The second phase is to train the RBF network with ELM. Compared with the RBF-ELM, the experimental results show that the proposed algorithm can improve the testing accuracy.

Keywords: Extreme learning machine · Radial basis function · Instance selection · Data condensation · Multi-scale

1 Introduction

As a simple and effective learning algorithm, extreme learning machine(ELM) [1] was proposed by Huang et al. in 2004. ELM is used for training single-hidden layer feed-forward neural networks(SLFNs). In ELM, the weights of input layer and biases of hidden nodes are randomly generated, and the weights of output layer are determined analytically. Huang et al. have proved in [2] and [3] that the ELM has universal approximation capability and good generalization performance. Since ELM was proposed in 2004, different variants of ELM have been proposed by different researchers. These variants improve the performance of ELM to some extent from different views. For example, kernel based ELM was proposed by Huang et al. for regression and multiclass classification [4]. Fully complex ELM [5] was proposed by Li et al. for solving equalization problems. Online sequential ELM [6] was proposed by Liang et al. for solving the sequential learning problems. In addition, in order to solve the problem of architecture selection of SLFNs trained with ELM, some algorithms have been proposed. For example, two incremental methods named I-ELM(Incremental-ELM) [7] and EM-ELM(Error Minimized-ELM) [8] were proposed in 2008 and 2009

respectively. Three pruning methods were proposed in 2008, 2010, and 2010, which are the P-ELM(Pruned-ELM) [9], OP-ELM(Optimally Pruned ELM) [10], and CS-ELM(Constructive hidden nodes Selection ELM) [11] respectively. RBF-ELM [12] is another kind of variant of ELM, which is tailored for radial basis function(RBF) networks. Similar to ELM, RBF-ELM also employs randomized method to initialize the centers and widths of RBF kernels, and analytically calculate the output weights of RBF networks. An excellent survey paper on ELM can be found in [13].

The key superiority of ELM [3] together with its various variants is that it needs no iterations, which dramatically reduces the computational time for training the model. But ELM and its variants also have some drawbacks, such as, (1) the predictive instability caused by randomly selecting the input weights and the hidden layer biases, i.e. ELM have variations in different trials of simulations; (2) For large data sets, the order of hidden layer output matrix H may be very high, which results in large memory requirement for calculating the Moore-Penrose generalized inverse of H . In RBF-ELM, because the two parameters are all generated by random approach, the problem (1) is still there. In this paper, we proposed a two-phase RBF-ELM learning algorithm. The first phase of the proposed algorithm is to select the centers of the RBF network rather than randomly initializing. The second phase is to train the RBF network with ELM. In other words, in our proposed method, only the width parameters are randomly generated, the center parameters are determined by instance selection method. Compared with the RBF-ELM, the experimental results show that the proposed algorithm can improve the testing accuracy. Due to the limitation of pages, we do not experimentally compare the difference of the predictive instability of the proposed method with RBF-ELM. We will investigate this problem in future works.

This paper is organized as follows. Section 2 provides preliminaries related to our work. Section 3 presents the method proposed in this paper. The experimental results are presented in Section 4. Section 5 concludes this paper.

2 Preliminaries

In this section, we briefly review the basic concepts and algorithms, including ELM and RBF-ELM.

2.1 ELM

ELM is a training algorithm for SLFNs, see Figure 1.

Given a training data set, $D = \{(x_i, t_i)\}$, $x_i \in R^d$, $t_i \in R^k$, $i = 1, 2, \dots, n$, where x_i is a $d \times 1$ input vector and t_i is a $k \times 1$ target vector, a SLFN with m hidden nodes is formulated as

$$f(x_i) = \sum_{j=1}^m \beta_j g(\omega_j \cdot x_i + b_j), (1 \leq i \leq n) \quad (1)$$

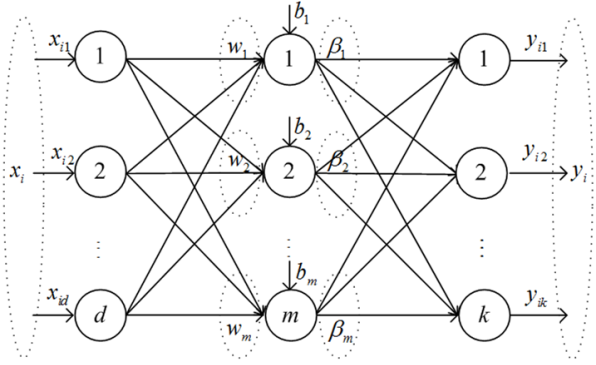


Fig. 1. Single-hidden layer feed-forward neural networks

where $\omega_j = (\omega_{j1}, \omega_{j2}, \dots, \omega_{jd})^T$ is the weight vector connecting the j th hidden node with the input nodes. b_j is the threshold of the j th hidden node. ω_j and b_j are randomly assigned. $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ is the weight vector connecting the j th hidden node with the output nodes. The parameters $\beta_j (j = 1, 2, \dots, m)$ may be estimated by least-square fitting with the given training data set D , i.e., satisfying

$$f(x_i) = \sum_{j=1}^m \beta_j g(\omega_j \cdot x_i + b_j) = t_i \tag{2}$$

Equation (2) can be written in a more compact format as

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

where

$$H = \begin{bmatrix} g(\omega_1 \cdot x_1 + b_1) & \cdots & g(\omega_m \cdot x_1 + b_m) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \cdot x_n + b_1) & \cdots & g(\omega_m \cdot x_n + b_m) \end{bmatrix} \tag{4}$$

$$\beta = (\beta_1^T, \beta_2^T, \dots, \beta_m^T)^T \tag{5}$$

and

$$T = (t_1^T, t_2^T, \dots, t_n^T)^T \tag{6}$$

H is the hidden layer output matrix of the network [1], where the j th column of H is the j th hidden node's output vector with respect to inputs x_1, x_2, \dots, x_n , and the i th row of H is the output vector of the hidden layer with respect to input x_i . If the number of hidden nodes is equal to the number of distinct training samples, the matrix H is square and invertible, and SLFNs can approximate these training samples with zero error. But generally, the number of hidden nodes is much less than the number of training samples. Therefore, H is a non-square matrix and one can not expect an exact solution of the system (3). Fortunately, it has been proved in [2] and [3] that SLFNs with random hidden nodes have the universal

approximation capability and the hidden nodes could be randomly generated. The least-square fitting is to solve the following equation:

$$\min_{\beta} \|H\beta - T\| . \tag{7}$$

The smallest norm least-squares solution of (7) can be easily obtained:

$$\hat{\beta} = H^\dagger T , \tag{8}$$

where H^\dagger is the Moore-Penrose generalized inverse of matrix H [14].

The ELM Algorithm [1] is presented in the following.

Algorithm: ELM

Input: Training data set $D = \{(x_i, t_i)\}$, an activation function g , and the number of hidden nodes m .

Output: The weights matrix β .

Step1: Randomly assign input weights ω_i and biases $b_j, j = 1, \dots, m$;

Step2: Calculate the hidden layer output matrix H ;

Step3: Calculate output weights matrix $\beta = H^\dagger T$.

2.2 RBF-ELM

RBF-ELM is a tailored ELM for training radial basis function networks (RBFNs), see Figure 2.

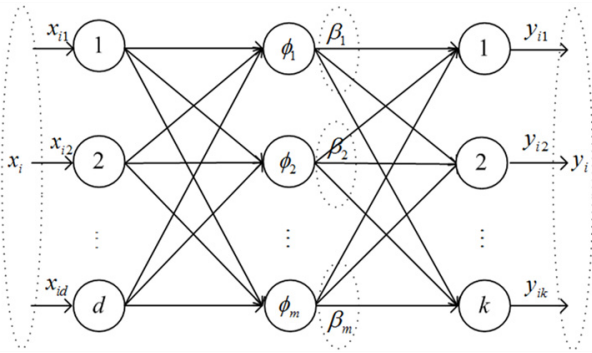


Fig. 2. Radial basis function networks

In Figure 2, $\phi_i(\cdot), i = 1, 2 \dots, m$ is kernel function, which is usually Gaussian:

$$\phi_i(x) = \phi(x, \mu_i, \sigma_i) = \exp\left(-\frac{\|x - \mu_i\|^2}{\sigma_i^2}\right) \tag{9}$$

where $\mu_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{id})^T$ is the i th kernel's center and σ_i is its impact width. $\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})^T$ is the weight vector connecting the i th kernel

and the output neurons. The output of a RBFN with m kernels for an input $x \in R^d$ is given by

$$f(x) = \sum_{i=1}^m \beta_i \phi_i(x) \tag{10}$$

Given a training data set, $D = \{(x_i, t_i)\}$, $x_i \in R^d$, $t_i \in R^k$, $i = 1, 2, \dots, n$. The RBFN shown in figure 2 can be modeled as

$$\sum_{i=1}^m \beta_i \phi(x_j) = o_j, j = 1, 2, \dots, n \tag{11}$$

Similar to SLFN case, the RBFN with m kernels can approximate these n samples with zero error means, so that

$$\sum_{j=1}^n \|o_j - t_j\| = 0 \tag{12}$$

That is to say there exists β_i , μ_i and σ_i such that

$$\sum_{i=1}^m \beta_i \exp\left(\frac{\|x - \mu_i\|^2}{\sigma^2}\right) = t_j, j = 1, 2, \dots, n \tag{13}$$

The equation (13) can be written compactly as

$$H\beta = T \tag{14}$$

where

$$H = \begin{bmatrix} \phi(x_1, \mu_1, \sigma_1) & \cdots & \phi(x_1, \mu_m, \sigma_m) \\ \vdots & \cdots & \vdots \\ \phi(x_n, \mu_n, \sigma_n) & \cdots & \phi(x_n, \mu_m, \sigma_m) \end{bmatrix} \tag{15}$$

$$\beta = (\beta_1^T, \beta_2^T, \dots, \beta_m^T)^T \tag{16}$$

and

$$T = (t_1^T, t_2^T, \dots, t_n^T)^T \tag{17}$$

Also similar to SLFN case, the least-square fitting of (14) is to solve the following equation.

$$\min_{\beta} \|H\beta - T\| \tag{18}$$

The smallest norm least-squares solution of (18) can be easily obtained:

$$\hat{\beta} = H^\dagger T \tag{19}$$

where H^\dagger is also the Moore-Penrose generalized inverse of matrix H .

The RBF-ELM Algorithm [12] is presented in the following.

Algorithm: RBF-ELM

Input: Training data set $D = \{(x_i, t_i)\}$, $i = 1, \dots, n$, an kernel function ϕ , and the number of kernels m .

Output: The weights matrix β .

Step1: Randomly assign kernel center μ_i and width σ_i , $i = 1, 2, \dots, m$;

Step2: Calculate the hidden layer output matrix H ;

Step3: Calculate output weights matrix $\beta = H^\dagger T$.

3 The Proposed Two-Phase RBF-ELM Algorithm

In this section, we present the proposed algorithm, which consist of two phases: the first phase is to determine the center of kernels with an instance selection algorithm based on multi-scale data condensation technique [15]. The second phase is to train the RBF network with ELM.

3.1 Selection Centers of the Kernels

In the proposed algorithm, only the width parameters are randomly generated, and the center parameters are determined with an instance selection algorithm, which is the multi-scale instance selection algorithm(MIS) proposed in [15]. In the dense regions, the scale is smaller, while in the sparse regions, the scale is bigger. The selected instances are used as the centers of the kernels. The instance selection algorithm is described as follows.

Algorithm: MIS

Input: Training data set $D = \{(x_i, t_i)\}$, $i = 1, \dots, n$, a positive integer k .

Output: The selected subset E of D .

Step1: Initialize $E = \emptyset$;

Step2: For each $x_i \in D$, find its k nearest neighbor in D , let N_i denote the set of data point x_i and its k nearest neighbors, and r_{k,x_i} denote the distance between x_i and its k nearest neighbor;

Step3: Calculate $x_j = \arg \min_{x_i} \{r_{k,x_i}\}$, $1 \leq i \leq n$;

Step4: Calculate $E = E \cup \{x_j\}$;

Step5: Calculate $D = D - N_i$;

Step6: Repeat Step2-Step5, until the $D = \emptyset$.

3.2 The Proposed Two-Phase RBF-ELM Algorithm

On the basis of the section 3.1, the proposed two-phase RBF-ELM(TP RBF-ELM) algorithm can be given in detail as follows.

Algorithm: TP RBF-ELM

Input: Training data set $D = \{(x_i, t_i)\}$, $i = 1, \dots, n$, an kernel function ϕ , and a positive integer k .

Output: The weights matrix β .

The first phase: determine the centers

Determine the centers of kernel with MIS algorithm, let the number of centers is m .

The second phase: train RBF network with ELM

Step1: Initialize the centers of kernel of RBF network with the m selected instances in the first phase, the number of hidden nodes of RBF network is m ;

Step2: Randomly assign impact width $\sigma_i, i = 1, \dots, m$;

Step3: Calculate the hidden layer output matrix H ;

Step4: Calculate output weights matrix $\beta = H^\dagger T$.

4 Experiments and Analysis

We experimentally compare the proposed algorithm with the original RBF-ELM algorithm on 7 UCI data sets [16] to verify the effectiveness of the proposed algorithm in two aspects, which are testing accuracy and the number of hidden nodes of the RBF network. The basic information of the 7 UCI data sets together with the size of training sets and testing sets are listed in Table 1. The experiments are conducted with MATLAB 7.1 on PC with 2.3GHz CPU and 4G RAM.

Table 1. The information of 7 UCI data sets selected for experiments

Data sets	#Attribute	#Class	#Instance	#Training	#Testing
Iris	4	3	150	100	50
Glass	10	7	214	110	104
Liver	6	2	345	170	175
Pima	8	2	768	576	192
Vehicle	18	4	846	420	426
Spambase	57	2	4601	3000	1601
Page Blocks	10	5	5473	2700	2773

The k is a predefined parameter, which determines the number of selected instances, in other words, it determines the number of hidden nodes of RBF network in the proposed algorithm. Finally, it determines the performance of the proposed algorithm. For different data set, the appropriate value of k is different. In this paper, we experimentally determine the optimal value of k . The relationship of the value of k and the testing accuracy is given in figure 3 and 4.

It can be seen from the figure 3 and 4 that for different data sets, the optimal value of k is really different, for example, for Iris data set, the optimal value of k is 27, while for Pima data set, the optimal value of k is 10. The experimental results on 7 UCI data sets are listed in Table 2. From the table 2, we can see that the proposed algorithm TP RBF-ELM is superior to the RBF-ELM in testing accuracy on all data sets. The experimental results are coincident with our guess. We are sure that the outperformance of the proposed algorithm is due to the contribution of elaborate selection of the centers of kernels in the first phase of the proposed algorithm.

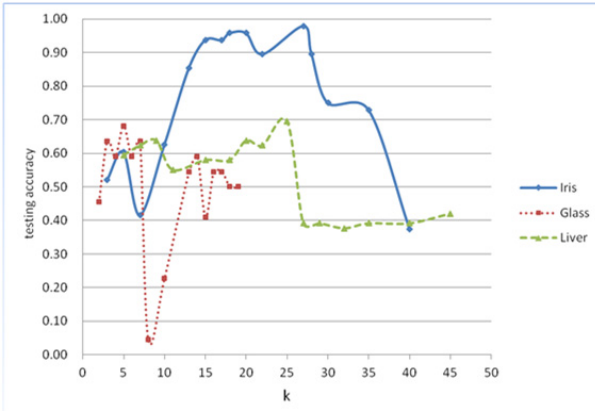


Fig. 3. The relationship of the value of k and the testing accuracy on 3 selected data sets

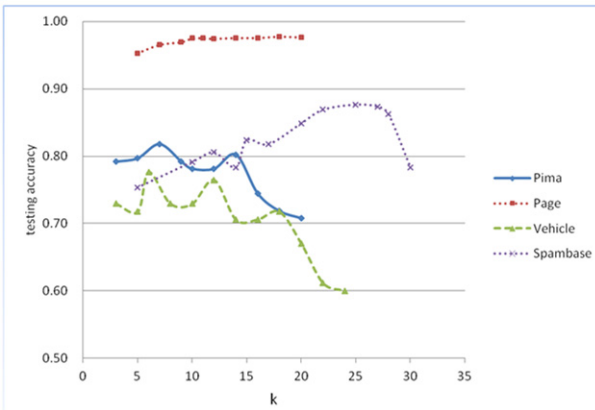


Fig. 4. The relationship of the value of k and the testing accuracy on 4 selected data sets

Table 2. The experimental results

Data sets	k	Testing Accuracy		#Hidden Nodes		
		RBF-ELM	TP RBF-ELM	RBF-ELM	TP RBF-ELM	RBF-ELM
Iris	27	0.95	0.98	17		25
Glass	5	0.65	0.68	30		60
Liver	25	0.68	0.70	30		15
Pima	10	0.76	0.79	30		35
Vehicle	6	0.78	0.80	120		117
Spambase	25	0.88	0.90	130		410
Page Blocks	10	0.95	0.97	34		160

5 Conclusions

This paper attempts to enhance the testing accuracy of RBF-ELM by the proposed algorithm, which is a two-phase RBF-ELM training algorithm. The first phase of the proposed algorithm elaborately selects the centers of kernels with an instance selection method. The second phase of the proposed algorithm is to train the RBF network with ELM. Because that only the width parameters are randomly generated, the proposed algorithm can improve the generalization ability, the experimental results compared with original RBF-ELM confirmed this conclusion. The experimental results verified that the proposed algorithm is effective and efficient.

Acknowledgments. This research is supported by the national natural science foundation of China (61170040, 71371063), by the natural science foundation of Hebei Province (F2013201110, F2013201220), by the key scientific research foundation of education department of Hebei Province (ZD20131028), and by the scientific research foundation of education department of Hebei Province (Z2012101).

References

1. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: A new learning scheme of feedforward neural networks. *IEEE IJCNN* **2**, 985–990 (2004)
2. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE T. Neural Network* **17**, 879–892 (2006)
3. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: Theory and applications. *Neurocomputing* **70**, 489–501 (2006)
4. Huang, G.B., Zhou, H.M., Ding, X.J., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE T. Syst. Man. Cy. B* **42**(2), 513–529 (2012)
5. Li, M.B., Huang, G.B., Saratchandran, P., Sundararajan, N.: Fully complex extreme learning machine. *Neurocomputing* **68**, 306–314 (2005)
6. Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N.: A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE T. Neural Network* **17**(6), 1411–1423 (2006)
7. Huang, G.B., Li, M.B., Chen, L., Siew, C.K.: Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing* **71**, 576–583 (2008)
8. Feng, G., Huang, G.B., Lin, Q., Gay, R.: Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE T. Neural Network* **20**, 1352–1357 (2009)
9. 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)
10. Miche, Y., Sorjamaa, A., Bas, P., Simula, O.: OP-ELM: Optimally pruned extreme learning mMachine. *IEEE T. Neural Network* **21**, 158–162 (2010)
11. Lan, Y., Soh, Y.C., Huang, G.B.: Constructive hidden nodes selection of extreme learning machine for regression. *Neurocomputing* **73**, 3191–3199 (2010)
12. Huang, G.B., Siew, C.K.: Extreme Learning Machine: RBF Network Case. *ICARCV* **2**, 1029–1036 (2004)

13. Huang, G.B., Wang, D.H., Lan, Y.: Extreme learning machines: A survey. *Int. J. Mach. Learn. Cyber.* **2**, 107–122 (2011)
14. Serre, D.: *Matrices: Theory and Applications*. Springer, New York (2002)
15. Mitra, P., Murthy, C.A., Pal, S.K.: Density-Based Multiscale Data Condensation. *IEEE T. Pattern Anal.* **24**(6), 734–747 (2002)
16. Blake, C., Merz, C.: UCI repository of machine learning databases in: <http://www.ics.uci.edu/mlearn/MLRepository.html>. Dept. Inf. Comp. Sc. UCI USA (1998)

Similarity and Decision Making

A Study on Decision Making by Thai Software House Companies in Choosing Computer Programming Languages

Vasin Chooprayoon^(✉)

Department of Information Technology, Faculty of Information Technology,
Rangsit University, Lak-hok, Muang, Pathumthani 12000, Thailand
vasin@rsu.ac.th

Abstract. Choosing good computer programming languages by software house companies can support software development. Managers of the companies have to make decision to choose the languages based on various criteria. This study aims to investigate the factors and criteria that influence the decision. The research findings show that characteristics of programmers, technology and tools, culture and society, problems occurring in software development process have high influence on the decision in choosing computer programming languages with statistical significance. Such information could be incorporated in a decision making system to optimize the appropriate language to be adopted.

Keywords: Computer programming languages · Thai software houses · Programming language selection · Programmer characteristics

1 Introduction

Computer programming languages are important tools for developing various applications, in particular as regard to their execution on various platforms and display on different medium. In the process of application development, programming languages need to be chosen as a main tool in fulfilling the development requirements. The software house companies have different criteria to choose an appropriate programming language, which is one of the most major decisions [1]. As application software has become essential in supporting various functions in an organization, many organizational functions can be operated automatically if the appropriate language and tools are chosen. In order to fulfill demand from the software customers, software house companies have to select appropriate programming language for developing purchase orders from customers in a timely and efficient manner.

1.1 Software Industry in Thailand

In Thailand, market value of software development and software services in 2012 is 34,481 million Baht increased from 2011 17.2% (29,418 million Baht) and software market is expanding continually [2].

Enterprise Software has the highest volume at 17,865 million Baht in 2011 and increased by 15.8% to 20,688 million Baht in 2012. Mobile Applications have the least production with only 1,065 million Baht in 2011 and it is expected to grow to 1,447 million Baht in 2012 at a growth rate of 35.9%. Software Services amount to 10,488 million Baht in 2011 and expect to expand to 12,346 million Baht in 2012 (17.7%) [2].

Framework of software industry promotion emphasizes reputation building for Thai software house companies worldwide, in order to enhance Thailand's successful reputation of Thailand's tourism industry, health industry, agricultural industry, etc. The Thai software sector tends to concentrate on the development of software to support the industries mentioned above. Cooperation of the software house companies becomes an important mechanism to develop large software solutions for highly efficient operations and to meet requirements of the industry sectors. Software is therefore viewed as a significant tool to enhance business strength and to improve economic value among the framework of software industry promotion 2012-2015.

Software Industry Promotion Agency (SIPA) defines five strategies for achieving success for Thai software industry. They are listed as follows [2]:

Strategy 1—develop reputation for Thai software entrepreneurs and push up them to the world stage

Strategy 2—promote excellence of Thai software industry

Strategy 3—contribute cooperation between software industry sector and other industries

Strategy 4—modify operational processes within the organizations

Strategy 5—contribute towards staff capabilities

1.2 Criteria for Selection of Computer Programming Languages

In general, choosing computer programming languages to be deployed by the software houses is based on technical characteristics such as ease of learning, ease of understanding, speed of development, help with enforcement of correct code, performance of compiled code, supported platform environments, portability, and fit-for-purpose [3]. Other criteria are secure programming practices, web applications development, web services design and composition, object oriented-based abstraction, reflection, aspect-orientation, functional programming, declarative programming, batch scripting, and user interface prototype design [4]. In addition, selecting a good language also consider factors such as whether the language: a) can run on any particular hardware or operating system, b) is standardized, and compiler implementations comply with this standard, c) supports software engineering technology, d) discourage/prohibit poor practices, and promote or supporting maintain activities, e) supports application domain(s) of interest, f) support requirements of system reliability and safety, g) compile implementations which are commensurate with the current state of technology, and h) is available for appropriate software engineering-based supporting tools and environments [5]. While previous studies have focused on technical criteria and characteristics of the languages, this study aims to find how different criteria related with computer language selection, and how the different criteria (factor) influence decision in using specific computer programming languages by software development companies located in the Kingdom of Thailand.

2 Research Design

This study is a quantitative research using questionnaires as research tools. Content and construct validity of the questionnaires were evaluated by five experts in programming languages, and proved by value of index of item objective congruence, IOC, which is closed to 1 ($\alpha = .9003$). Research population is based on 340 software companies registered as members of The Association of Thai Software Industry (ATSI) [6].

2.1 Research Variables

This study covers three variables: a) stimulus variables (predictors) which are programmer characteristics, related technology and tools, culture and society, and problems occurring in software development process; b) dependent variables which is decision making of software house companies to use computer programming languages; and c) criterion variable which is forms of software development which is composed of two variables: a) original development and b) import from abroad and modify for Thai demand. These variables are incorporated in the research model as shown in Figure 1.

2.2 Research Model

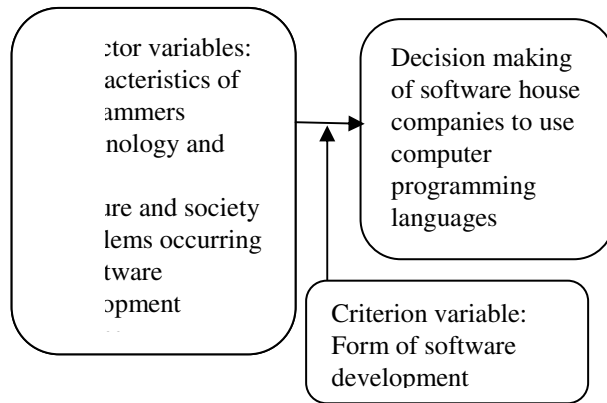


Fig. 1. Research model

The variables in Figure 1 are divided into fifty-two sub-variables as shown in Table 1.

2.3 Data Processing

The multiple regression statistics were used to calculate data from returned questionnaires. Assumptions for multiple regression are 1) stimulus variables have no internal relationship with each other (autocorrelation), 2) values of prediction errors have normal distribution, 3) prediction errors have zero mean (\bar{X}), and 4) prediction errors have variance constant.

Table 1. Characteristics of programmers

Characteristics of Programmers
1) Profession expertise and specialization (CP1)
2) Ability to communicate in English (CP2)
3) Initiating creativity (CP3)
4) Team working (CP4)
5) Ability to identify and explain difficult levels of tasks (CP5)
6) Ability to identify and explain how to gain more benefits from embedded languages (CP6)
7) Ability to choose appropriate computer language for software development (CP7)
8) Perceiving how to respond pressures in task areas (CP8)
9) Understanding customer needs (CP9)
10) Perceiving how to receive promotion and contribution in working path (CP10)
11) Comprehension of how to develop existing potentials of colleagues (CP11)
12) Ability to prioritize job functions (CP12)
13) Comprehension of how to extract maximum potentials of colleagues for beneficial fulfillment (CP13)
14) Ability to analyze problems in team working (CP14)
15) Ability to handle boring tasks (CP15)
16) Ability to ask for collaboration from other units for completing projects (CP16)
17) Ability to develop operating patterns for scalability of tasks and new various tasks (CP17)
18) Ability to communicate with colleagues effectively (CP18)
19) Ability to compromise with conflict team workers (CP19)
20) Ability to handle protracted problems in team (CP20)
21) Ability to handle chaos and disorder in organization (CP21)
Technology and Tools
1) Necessary criterion for choosing computer language used in software development (TT1)
2) Using advanced computer languages which have special command sets and syntaxes (TT2)
3) Long-term searching by specialists in order to find the best computer languages for software development (TT3)
4) Various straight and weakness of each computer language (TT4)
5) Choosing computer language against types of computers, programming, and programmer specialization(TT5)
6) computer languages and matching quality of software products develop by software houses (TT6)
7) Linkage between computer languages and pricing software products (TT7)
8) Linkage between complex computer languages and building trust for software products (TT8)
Impact of Culture and Society
1) Current studying and teaching patterns (CS1)
2) Educational software is a part of reinforcement of people who get fewer education opportunities (CS2)
3) Investment on educational software development (CS3)
4) Decreasing and increasing a number of employees working in software development projects (CS4)
5) Presentation strategies for promoting new software products (CS5)
6) Fore planning for software development (CS6)
7) Certain procedures and standard of working process in software development (CS7)
8) Avoidance of risk with high responsibilities of employees including all process monitoring and tracking (CS8)
9) Determining duration of software development cycle including retrospective evaluation (CS9)
10) Employees satisfaction to working tasks including relationship between employees an software house (CS10)

Table 1. (Continued.)

Problems Occurring in the Software Development Process
1) Programmers are dominant customers’ needs (P1)
2) Customers’ needs (P2)
3) New version of computer languages (P3)
4) Duration of software development (P4)
5) Difficult to upgrade computer languages (P5)
6) There are many teams working in the same project (P6)
7) Organizing team as many sub-modules (P7)
8) Using too advanced techniques in software development process (P8)
9) Organization culture (P9)
10) Over focus on software system design (P10)
11) Conflicts between new codes and legacy codes (P11)
12) Coding is different from fore commitment (P12)
13) Strong feeling of anxiety in searching solutions during software development (P13)

3 Research Findings

The results generated from multiple regression analysis showed that all factors (stimulus variables) have no autocorrelation (*internal relationship of stimulus variable values*). The Durbin–Watson statistic detected the autocorrelation in the prediction errors (residuals) [7] resulted the Durbin-Watson value at 2.424 which is in the range 1.5 to 2.5 means that there is no autocorrelation (Table 2).

Table 2. Durbin-watson test

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.880	.775	.699	.21253	2.424

The normal distribution of prediction errors was validated by Normal Prob. Plot displayed in the Figure 2. The small circles are Prob. values scattering near the straight line means that there are normal distributions of the prediction error values.

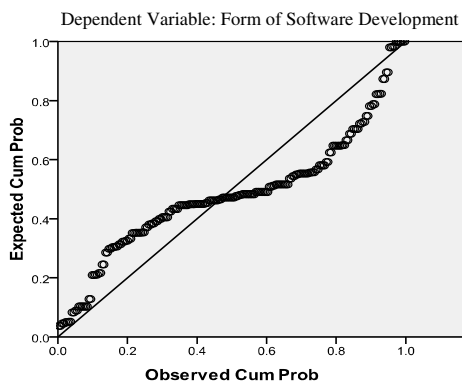


Fig. 2. Normal P-P plot of regression standardized residual

Table 3 proved mean (\bar{X}) of residual equals .000 which follows the multiple regression assumption.

Table 3. Residuals statistics

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	.7324	2.2860	1.1827	.34090	208
Residual	-.37884	.74115	.00000	.18391	208
Std. Predicted Value	-1.321	3.236	.000	1.000	208
Std. Residual	-1.782	3.487	.000	.865	208

Figure 3 displays variance values of prediction errors scattering above and under level 0.0 of regression standardized residual in equal area that means the errors have variance constant.

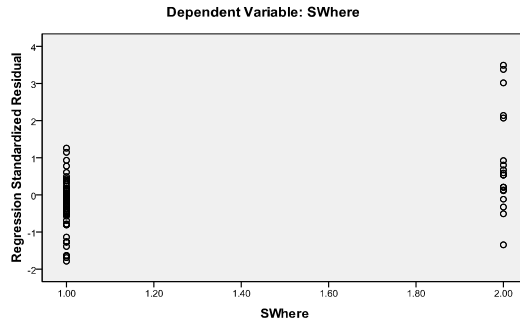


Fig. 3. Scatter plot of regression standardized residual

The detection of the above assumptions has proved the regression analysis by using the enter method. Table 4 showed that 52 factors have high influence to Thai software house companies’ decision to use computer programming languages. The multiple correlation coefficients (R) is .880 which is able to predict influence of the 52 factors to the decision at 77.5% (coefficient of determination or R² is .775) with statistical significance <.001 (p-value=.000) and standard error of estimation ±.212.

By considering multiple regression coefficient values, we found that factor CS8 can predict the decision most highly (β=2.334) followed by factor TT6, TT5, CS5, CS1, and CP14 (β=2.157, 2.188, 1.450, 1.151, and 1.055 in ascending order).

The decision equation to use computer programming languages derived from entering raw scores of all factors into the equation is shown as follows:

$$\begin{aligned}
 \widehat{\text{Decision}} = &-.068\text{CP}_1 - .146\text{CP}_2 + .096\text{CP}_3 - .505\text{CP}_4 - \\
 &.165\text{CP}_5 - .010\text{CP}_6 + .102\text{CP}_7 - .161\text{CP}_8 + .174\text{CP}_9 - \\
 &.069\text{CP}_{10} + .314\text{CP}_{11} + .090\text{CP}_{12} - .320\text{CP}_{13} + .467\text{CP}_{14} + \\
 &.122\text{CP}_{15} - .343\text{CP}_{16} + .230\text{CP}_{17} - .200\text{CP}_{18} + .394\text{CP}_{19} + \\
 &.129\text{CP}_{20} - .014\text{CP}_{21} + .076\text{TT}_1 + .459\text{TT}_2 - .480\text{TT}_3 + \\
 &.128\text{TT}_4 + .858\text{TT}_5 - .886\text{TT}_6 + .061\text{TT}_7 + .480\text{TT}_8 + \\
 &312\text{CS}_1 - .191\text{CS}_2 - .407\text{CS}_3 - .057\text{CS}_4 + .573\text{CS}_5 - \\
 &.044\text{CS}_6 + .482\text{CS}_7 - 1.249\text{CS}_8 + .343\text{CS}_9 + .201\text{CS}_{10} - \\
 &.241\text{P}_1 - .026\text{P}_2 - .369\text{P}_3 - .129\text{P}_4 + .365\text{P}_5 + .089\text{P}_6 - \\
 &.161\text{P}_7 + .074\text{P}_8 + .147\text{P}_9 + .274\text{P}_{10} + .038\text{P}_{11} + .088\text{P}_{12} - \\
 &.147\text{P}_{13}
 \end{aligned}
 \tag{1}$$

Table 4. Multiple regression for predicting factors influencing decision to use computer programming languages from

Variables	b	SE	β	t	p-value
CP1	-.068	.054	-.146	-1.268	.207
CP2	-.146	.071	-.331	-2.056	.041*
CP3	.096	.064	.241	1.520	.131
CP4	-.505	.150	-1.158	-3.360	.001**
CP5	-.165	.087	-.319	-1.901	.059
CP6	-.010	.083	-.021	-.124	.901
CP7	.102	.079	.225	1.288	.200
CP8	-.161	.096	-.277	-1.668	.097
CP9	.174	.085	.322	2.061	.041*
CP10	-.069	.081	-.110	-.852	.395
CP11	.314	.109	.665	2.873	.005*
CP12	.090	.131	.148	.685	.495
CP13	-.320	.081	-.890	-3.972	.000**
CP14	.467	.090	1.055	5.192	.000**
CP15	.122	.063	.233	1.925	.056
CP16	-.343	.148	-.656	-2.312	.022*
CP17	.230	.065	.487	3.556	.000**
CP18	-.200	.115	-.470	-1.739	.084
CP19	.394	.158	.938	2.494	.014*
CP20	.129	.179	.263	.722	.471
CP21	-.014	.039	-.029	-.355	.723
TT1	.076	.082	.159	.918	.360
TT2	.459	.153	.806	3.000	.003*
TT3	-.480	.154	-1.044	-3.111	.002*
TT4	.128	.148	.169	.865	.389
TT5	.858	.157	2.188	5.455	.000**
TT6	-.886	.197	-2.157	-4.497	.000**
TT7	.061	.154	.139	.394	.694
TT8	.480	.134	1.394	3.577	.000**
CS1	-.312	.091	-1.151	-3.441	.001**
CS2	-.191	.119	-.279	-1.611	.109
CS3	-.407	.078	-.737	-5.224	.000**
CS4	-.057	.078	-.124	-.730	.467
CS5	.573	.087	1.450	6.604	.000**
CS6	-.044	.043	-.099	-1.019	.310
CS7	.482	.198	.630	2.432	.016*
CS8	-1.249	.163	-2.334	-7.649	.000**
CS9	.343	.096	.808	3.557	.000**
CS10	.201	.111	.387	1.817	.071
P1	-.241	.076	-.561	-3.177	.002*
P2	-.026	.071	-.052	-.368	.713
P3	-.369	.108	-.592	-3.405	.001**
P4	-.129	.147	-.415	-.877	.382
P5	.365	.156	.770	2.344	.020*
P6	.089	.077	.209	1.153	.251
P7	-.161	.095	-.512	-1.702	.091
P8	.074	.137	.142	.539	.591
P9	.147	.096	.278	1.528	.129
P10	.274	.085	.558	3.214	.002*
P11	.038	.073	.095	.524	.601
P12	.088	.116	.159	.754	.452
P13	-.147	.074	-.239	-1.974	.050*

Constant = .543; SE_{est} = ±.212

R = .880; R² = .775; F_{52,155} = 10.242; p-value = .000

*Statistical significance at 0.05

** Statistical significance at 0.001

4 Conclusions

The research findings imply that programmer characteristics, related technology and tools, culture and society, and problems occurring in software development process have high influence on the decision to choose computer programming languages of the software house companies in Thailand. It is significant to make decision to choose the languages further than the technical characteristics of the languages. It may made the software companies more able to make right decisions. Providing appropriate languages can develop customers' required software applications; and hold up cost-effective in software development process, supply valuable, and maintain reliable business. Knowledge from this study could be incorporated into a decision making system to optimize the appropriate language to be adopted thereby contributing towards the software industry in the Kingdom of Thailand.

Acknowledgements. This research is supported by the Faculty of Information Technology, Rangsit University, Thailand. Special thanks to Niti Jitwattanatam for data gathering from the software companies.

References

1. TSG Library of Knowledge, Choosing your programming language (2013). http://library.techguy.org/wiki/Choosing_your_programming_language (cited January 3, 2013)
2. Software Industry Promotion Agency (SIPA), Strategic Plan for Software Industry Promotion 2012-2015, Ministry of Information Communication and Technology, Bangkok (2013)
3. Britton, C.: Choosing a Programming Language (2008). <http://msdn.microsoft.com/en-us/library/cc168615.aspx> (cited March 10, 2013)
4. Al-Qahtani, S.S., et al.: Comparing Selected Criteria of Programming Languages Java, PHP, C++, Perl, Haskell, AspectJ, Ruby, COBOL, Bash Scripts and Scheme Revision 1.0 (2010)
5. Lawlis, P.K.: Guidelines for Choosing A Computer Language: Support For The Visionary Organization (1997). <http://archive.adaic.com/docs/reports/lawlis/content.htm> (cited March 7, 2013)
6. The Association of Thai Software Industry (ATSI) (2013). <http://www.atsi.or.th> (cited March 1, 2012)
7. Durbin, J., Watson, G.: Testing for Serial Correlation in Least Squares Regression. I. *Biometrika* **37**, 409–428 (1950)

An Improved Method for Semantic Similarity Calculation Based on Stop-Words

Haodi Li^(✉), Qingcai Chen, and Xiaolong Wang

Intelligent Computing Research Center, Computer Science and Technology,
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
1hd911107@gmail.com

Abstract. Text similarity calculation has become one of the key issues of many applications such as information retrieval, semantic disambiguation, automatic question answering. There are increasing needs of similarity calculations in different levels, e.g. characters, vocabularies, syntactic structures and semantic etc. Most of existing semantic similarity algorithms can be categorized into statistical based methods, rule based methods and combination of these two methods. Statistical methods use knowledge bases to incorporate more comprehensive knowledge and have the capability of reducing knowledge noise. So they are able to obtain better performance. Nevertheless, for the unbalanced distribution of different items in the knowledge base, semantic similarity calculation performance for low-frequency words is usually poor. In this work, based on the distributions of stop-words, we proposes a weights normalization method for semantic dimensions. The proposed method uses the semantic independence of stop-words to avoid semantic bias of corpus in statistical methods. It further improves the accuracy of semantic similarity computation. Experiments compared with several existing algorithms show the effectiveness of the proposed method.

Keywords: Semantic similarity · ESA · Stop-words · Semantic dimension normalization

1 Introduction

Classical methods of computing the word level semantic similarity can be categorized into two categories: rule-based methods and statistical methods. The rule-based methods use dictionary or ontology knowledge base to build a tree (or a graph) structure for the calculation of similarities. The statistical based methods are further divided into methods of using or not using a knowledge base. Statistical methods that make use of knowledge bases usually achieve better results because of its ignorance to the noisy information in the knowledge base [1]. However, the uneven distributions of the knowledge base on different subjects greatly affect on the statistical weights of word-based semantic dimensions, which usually causes the poor performance of statistical methods for low-frequency words. To address this issue, there are some methods that combined rule based and statistical methods. They have obtained certain good results, but have not solved the problem fundamentally.

This paper presents a method that reduces the requirements of the size and the structure of knowledge base for statistical methods. It improves the final result of semantic similarity calculation of clauses. In a sentence, stop-words have no practical significant meaning and thus are semantically irrelevant features in statistical distribution. For example, "is" is a stop-word and its probability distribution is independent from the meaningful notional word "methods". The semantic independency of stop-words in a knowledge base helps to avoid the unbalance of semantic dimension in common statistical methods.

In semantic similarity calculation, rule-based methods generally use the ontology knowledge. The semantic similarity is computed by using the ontology structure and properties between the ontology items. At present, most of the rule-based algorithms employ WordNet [2] as its ontology dictionary. There are many mature methods that are based on WordNet, e.g., the OHIC that is based on the concept of local density, the amount of information and the depth[3], and the redefined density and depth based method[4]. However, just as in the WordNet, the connection between word concepts need to be artificially built. Both the number of concepts and their connections are limited. Thus they are usually not reliable enough for practical applications. On the other hand, the algorithms based on the paths of structures constructed for large-scale ontology bases, e.g. the Wikipedia, are able to get higher recall rate. The typical algorithms are Wiki Relate algorithm[5] and WLM algorithm[6]. The main issue for this kind of algorithms is that they are more easily affected by noisy data. It is because that the data structure of the Wikipedia is not as strong as the ontology dictionary. Therefore, the precisions of the methods using the ontology structure constructed from Wikipedia are not very good.

Another type of semantic similarity calculation algorithms is statistical methods. One type of statistical methods, e.g., the LSA, do not require artificial building of knowledge base, but their performance are depended on the initial training set. For the statistical algorithms based on the statistic of knowledge base, they need the support of larger knowledge base to get satisfied performance. For example, explicit semantic analysis (ESA) algorithm[7] has achieved the state-of-art performance by selecting the Wikipedia as knowledge base. The main idea of ESA algorithm is that the natural concept (or called "concept" for short) is the semantic units that are understandable and explainable for human beings, and a text can be made up by these concepts given with different weights. In ESA, each knowledge entry (corresponding to an article) in Wikipedia is a natural concept. For words contained in each of the knowledge entry, an inverted index of vocabulary-entry (Word-Concept) is constructed. The corresponding value of the inverted index is the weight of the word for the corresponding entry. By this way, for a given lexicon, a weight vector is constructed for each knowledge entry. The dimension of the weight vector is the size of the lexicon. Usually, the TF-IDF of a word is used as the weight value. To calculate the semantic similarity, ESA algorithm firstly maps each knowledge entry into the vector space. The cosine distance is then applied to calculate semantic similarity between knowledge entries. This algorithm is called explicit semantic analysis since it directly computes the similarity of two entries via the vector space model. As a statistical based algorithm, ESA algorithm overcomes the drawbacks from of traditional methods based on the structure of knowledge base such as WordNet and Wikipedia.

2 Analysis of ESA Method

Wikipedia is a multilingual, mass and constantly updated knowledge base, and ESA algorithm can effectively use it for semantic similarity calculation. The algorithm which utilizes the form of a unified computing similarity between words, sentences and texts, has higher practicability. But there are still some issues to overcome:

First, for each article, ESA algorithm treats Wikipedia entries as a high-dimensional vector, which approaches the potential semantic if the basis dimensions of the vector space are linearly independent. However, some concepts such as "machine learning" and "data mining" on the semantics of these two items are obvious not independent. In fact, the method based on the path of Wikipedia knowledge base entries is to calculate the similar relations via the semantic similarity between entries. Furthermore, collection of knowledge base for different categories of knowledge items are not balanced and the number of some categories of knowledge items are even overmuch, so the calculation of statistical based algorithms using ESA algorithm will be bias to part of the categories or semantic dimensions contain fewer concepts. The unbalance or bias existed in semantic dimensions. In this paper we called it semantic dimension offset.

As a result of semantic dimension offset, ESA algorithm does not use the same dimension benchmark in computing similarity for multiple sets that have completely different words, which causes that their similarity values may not be comparable. Our experiment shows that ESA algorithm is able to get comparable results of entry similarities that are calculated for a given entry with all other entries.

As shown in Table 1, because of the semantic dimension offset, different words that are puny difference in artificial discriminant values, may be discrepancy in the ESA algorithm calculation results (e.g., the 1st and 2nd rows in Table 1). With the same word in calculation, the similarity values of ESA algorithm are more consistent to artificial discriminant values (e.g., the 2nd and 3rd rows). Nevertheless, if the semantic dimension offset is serious, even for the same word, its ESA calculated similarity values with other words are also inconsistent with artificial discriminant values (e.g., 5th and 6th rows in Table 1).

Second, ESA algorithm is based on statistics algorithm of Wikipedia knowledge base, so it has more requirements for the completeness of employed knowledge base. Wikipedia is artificial and complicated, thus mass of low frequency vocabulary is inevitable because of long tail effectiveness.

In low frequency words set, the results of calculating semantic similarity between these words will not only unstable due to semantic dimension offset, but also unreliable due to the insufficient samples.

3 Improved Method Based on Semantic Independent

3.1 Classification of Stop-Words

Stop words generally can be divided into two types. One type is frequently used function words existed in the natural language, such as "that", "is" and "on" in English, "是" (means "yes") in Chinese, etc. Compared with content words, function words have

much less significance in representing meaning. Another type of stop words is the extremely frequent content words. This kind of words has the solid semantic information, such as "want" and "through", but many of them are treated as noise data and are also included in the stop-words list because of their extremely high frequency in the corpus.

Table 1. The bias of ESA semantic dimension

W1	W2	Human rate	ESA rate
tiger	jaguar	8.00	0.026
stock	market	8.08	0.172
stock	phone	1.62	0.018
stock	CD	1.31	0.005
jaguar	cat	7.42	0.024
jaguar	car	7.27	0.103
tiger	jaguar	8.00	0.026

This paper assumes the first type of stop words because there are no actual semantic meanings for them. Therefore, in the text, their distributions are only dependent on the scale of the corpus, and are independent from the semantic distributions.

3.2 Semantic -Independent of Stop Words

In order to verify the correctness of the hypothesis, we choose the standard text classification corpus Reuters. The categories contains more than ten thousand words are chosen for statistic. The distributions for stop word "the" and the content word "money" in different categories are shown in figure 1.a and figure 1.b respectively. The horizontal axis of Figure 1 represents different categories that are labeled by the first two letters of their category names. The vertical axis is for the probability distribution of the stop words "stop" and "word" in the copra of corresponding categories. From this figures it can be seen that probabilities of stop word "the" is higher than the content word "money", and the distribution of stop word in each category is relatively stable, while the distribution of the content word is obvious dependent on categories.

3.3 Normalize Semantic Dimension Based on Stop Words

Based on the semantic-independent property of stop words, this section proposes a semantic dimension normalization method. It firstly calculates the similarity of each content word with every element in a subset of stop words by ESA method on Wikipedia corpus. Then the average similarity value is taken over the stop words and is used as the normalization factor of the given content word. By the ESA method, the calculating formula for the similarity of a given content word x with a stop word sw is

$$Vec_x = \cos \langle \overline{r_{sw}}, \overline{r_x} \rangle \quad (1)$$

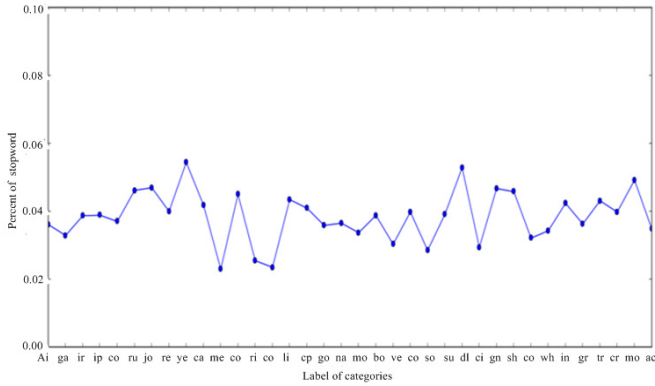


Fig. 1a. Distributions of stop word “stop” by category

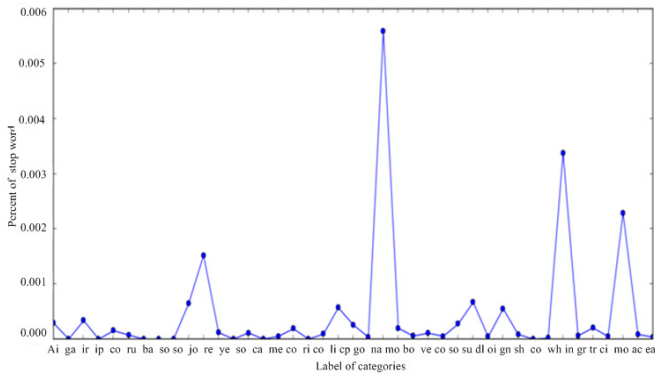


Fig. 1b. Distributions of content word “money” by category

For example, suppose that for a content word "retrieval", its projection vector \vec{r}_a in the vector space is $\langle 0, \dots, 0.5, 0, 0.7, 1.1, \dots \rangle$, and the matching of stop words $\langle \dots, 0.4, 0, 0.3, 0.9, \dots \rangle$. The normalization factor is $1.0025 / |\vec{r}_{sw}|$.

Since big differences among the weights of semantic dimensions for different words cause the bias of computing and comparing the similarity values. To correct the inconsistency caused by unbalanced word vector distributions, we use the equation(2) to calculate the normalized similarity:

$$Sim_{con}(W_1, W_2) = ESA(W_1, W_2) \cdot \sum_{i=1}^2 \frac{\alpha}{\alpha + ESA(W_i, W_{sw})} \tag{2}$$

Here W_1 and W_2 are two words for calculating the semantic similarity. $ESA(W_i, W_{sw})$ is the normalization factor Vec_i ; α is a smoothing parameter top reserving the valid of the equation in case that a normalization factor gets zero value.

3.4 Linearization

The statistical similarity calculation method based on knowledge base ESA algorithm is to map words into the semantic vector space, and then using the cosine similarity to compute the similarity values of the two vectors. Because the cosine function is not a linear function, the similarity distribution of ESA algorithm is not linear. For example, the similarities of human discriminants for word pairs "stock" and "stock", "stock" and "market", "stock" and "egg" are 10, 8.08, 1.81, respectively, and are 1, 0.808, 0.181 after normalization. However, calculation results by using ESA are 1, 0.17, and 0.0069 respectively.

Although the results of ranking are still kept in consistent, and using the ESA algorithm to calculate the semantic similarity does not adversely affect on the ranking results, its distribution is quite different from human's judgments. This distribution difference will greatly affect on the counting of absolute similarity value. Therefore, it is necessary to make the ESA similarity calculation results to be linear.

Because that it is difficult to estimate the real distribution of semantic similarities on large-scale corpus, this paper directly uses the approximate linearization formula (3) as follows:

$$Sim_{linear} = (2Sim_{esa} - Sim_{esa}^2)^\theta, 0 < \theta \leq \frac{1}{2} \quad (3)$$

When the value of θ is 0.3, after linearization, the ESA results in the example above become 1, 0.7045 and 0.2764 respectively, which are more comparable with the results of artificial judging semantic similarities.

4 Experiments

Statistical methods of ESA in this paper uses Wikipedia texts, in both Chinese and English article database, as training corpus. We extract more than 3 million entries in English, and 0.5 million Chinese entries (short entries had been filtered);

The test corpus of lexical level similarity calculation are Word Similarity-353[8](WS353) and Words-240[9][10] (WD240). Each data set has multiple records and each record consists of a pair of vocabulary and artificial judging semantic similarity value.

4.1 Results of Improved Methods

We use the standard TF-IDF algorithm as the feature extraction method in ESA. The experiments set smoothing parameter α to 0.1, and using the Spearman correlation coefficient to compare the algorithm calculated results and the results of artificial building.

Table 2 shows the effect semantic dimension bias on the similarity calculation results. The average dimension bias rate is the average value of dimension bias that are calculated by formula 1 for two words and the error rate is calculated by formula 4 as following. Here the variable H_{rate} is the Human rate, and Sim_{linear} is calculated by formula 3. The λ is set to 0.1 in this case. It can evaluate the discrepancy between a single calculated value and its parallel standard value.

$$E_{rate} = \left| \lambda H_{rate} - Sim_{linear} \right| \quad (4)$$

Table 2. Error rate and dimension bias rate

W1	W2	Error rate	Average dimension bias rate
tiger	jaguar	0.389	0.005
stock	market	0.101	0.012
stock	phone	0.205	0.009
stock	CD	0.120	0.013
jaguar	cat	0.341	0.007
jaguar	car	0.129	0.019

Table 2 shows that the dimension bias rate is correlation with error rate. The Error rate increased significantly when the bias rate falls below 0.01.

Full test results are shown in table 3. The Spearman ranks correlation coefficients of LIU, ICSUBCATEGORYNODE, WLM, WLT, ESA and ESA with dimension normalized(NESA) methods are shown in table 3. The test dataset are WS353 and WD240,

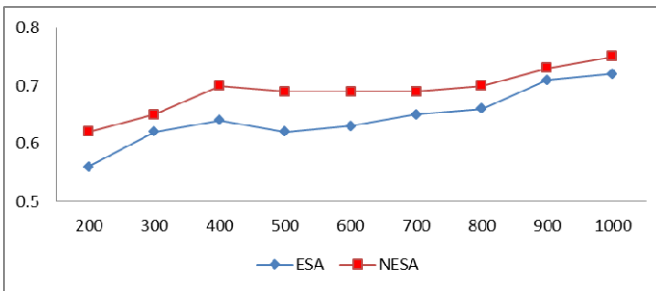
From the result, we can find that the NESA method based on consistent semantic dimension of stop-words can effectively enhance the performance of semantic similarity calculation. It reduces the effect of unbalanced distribution of words in the knowledge base on statistical similarity computing methods. Since English Wikipedia is more comprehensive than Chinese Wikipedia. Caused by the size limitation of knowledge base, the effect of the experimental dataset in English (WS353) is better than in Chinese data set (WD240)

4.2 Results for Low-Frequency Words

We also compared the similarities with the low-frequency words on WS353 dataset to evaluate the effectiveness of the proposed method. As shown in figure 2, vertical axis corresponds to the values of Spearman phase relations. And each horizontal point corresponding to the words in which the number of non-zero values in its word vector is not greater than the labeled horizontal axis value. The correlation coefficients show that the effectiveness of dimension normalization on the words of non-zero dimensions below 1000 is more obvious than ESA algorithm.

Table 3. Correlation values in WS353 and WD240

Methods	Correlation	
	WS353	WD240
LIU	0.4202	0.3351
WU	0.3205	0.2793
ICSubCategoryNodes	0.2803	0.2093
WLT	0.5126	0.4160
ESA	0.7448	0.5427
NESA	0.7644	0.5715

**Fig. 2.** Improve in low-frequency words

Experiments in this section show that the proposed method, to a certain extent, can reduce the dependency of statistical algorithm on the scale of knowledge base corpus. The word level performance is effectively improved for the semantic similarity calculation of low dimension words. But due to the inherent nature of statistical algorithm, the accuracy of calculation is still related with dimension falls. The corpus of knowledge base on statistical algorithm is still need a considerable scale.

5 Conclusions

This paper proposes an improved semantic similarity calculation method based on stop-words by using distribution independence of stop-words with semantic features. In addition to all eviated corpora disequilibrium bias in the semantic dimension of knowledge bases in a certain extent. Experimental results show that the proposed method improves the semantic similarity algorithm based on statistical calculation results effectively, especially for low frequency words. The proposed method also reduces the dependency on the scale of corpus.

The next research direction is to use independent features of stop-words to build knowledge base with non-bias dimension. This reduces the complexity of the semantic similarity calculation and overcomes the influence of the corpus size of knowledge base for statistical based algorithms. Therefore, it improves the effectiveness of semantic similarity computation.

Acknowledgements. This paper is supported in part by grants: Science and technology research and development funds of Shenzhen (JC201005260175A), NSFCs (National Natural Science Foundation of China) (61173075 and 61272383), Strategic Emerging Industry Development Special Funds of Shenzhen (ZDSY20120613125401420 and JCYJ20120613151940045) and Key Basic Research Foundation of Shenzhen (JC201005260118A).

References

1. Kang, C., Yu, Z.: Combination of semantic similarity and Hidden Markov Model in Chinese question classification. *Journal of Computational Information System* **8**(3), 1031–1038 (2012)
2. Buscaldi, D., Le, R.J., Flores, J.J.G., et al.: LIPN-CORE: Semantic Text Similarity using n-grams, WordNet, Syntactic Analysis, ESA and Information Retrieval based Features. In: *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pp. 162–168 (2013)
3. Bin, S., Liying, F., Jianzhuo, Y., et al.: Ontology-Based Measure of Semantic Similarity between Concepts. *WRI World Congress on Software Engineering* **2**(4), 109–112 (2009)
4. Wang, T., Hirst, G.: Refining the notions of depth and density in WordNet-based semantic similarity measures. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1003–1011 (2011)
5. Strube, Michael, and Simone Paolo Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. vol. **6**, pp. 1419–1424. *AAAI* (2006)
6. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: *Proceeding of AAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pp. 25–30. *AAAI Press* (2008)
7. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. *IJCAI* **7**, 1606–1611 (2007)
8. Finkelstein, L., Gabrilovich, E., Matias, Y., et al.: Placing search in context: The conceptrevisited. *ACM Transactions on Information Systems* **20**(1), 116–131 (2002)
9. Xiang, W., Yan, J., Bin, Z.: Computing Semantic Relatedness Using Chinese Wikipedia Links and Taxnomy. *Journal of Chinese Computer Systems* **32**(11), 2237–2242 (2011)
10. Li, H., Chen, Q., Wang, X.: A Combined Measure for Text Semantic Similarity. In: *Proceedings of the 2013 International Conference on Machine Learning and Cybernetics*. vol. 4, pp. 1869–1873 (2013)

Learning with Uncertainty

Sensitivity Analysis of Radial-Basis Function Neural Network due to the Errors of the I.I.D Input

Jie Li^{1(✉)}, Jun Li², and Ying Liu¹

¹Faculty of Mathematics and Computer Science, Hebei University,
Baoding 071002, China
lijiehb@gmail.com

²Department of Computer Science, Communication Training Base of General Staff,
Xuanhua 075100, China

Abstract. An important issue, in the design and implementation of a Radial-Basis Function Neural Network (RBFNN), is the sensitivity of its output to input perturbations. Based on the central limit theorem, this paper proposes a method to compute the sensitivity of the RBFNN due to the errors of the inputs of the networks. For simplicity and practicality, all inputs are assumed to be independent and identically distributed (i.i.d.) with uniform distribution on interval (a, b) . A number of simulations are conducted and the good agreement between the experimental results and the theoretical results verifies the reliability and feasibility of the proposed method. With this method, not only the relationship among the sensitivity of RBFNN, input error ratios and the number of the neurons of the input layer but also the relationship among the sensitivity, input error ratios and the number of the neurons of the hidden layer is founded.

Keywords: Radial basis function (RBF) · Neural networks · Sensitivity analysis

1 Introduction

Sensitivity is a measure to evaluate the degree of the response of a network due to the error of its input and others parameters. So sensitivity analysis is a hot issue of networks research. The major purpose of this paper is to study the error of the output by comparing the unperturbed case and the perturbed case. In the past decade, a lot of methods are proposed to study the sensitivity and applications are pointed out to direct us to design the networks. For example, Stevenson et al. 1 use a hypersphere to make a mathematical model to analyze the sensitivity of Madalines and derived the probability of erroneous output, which is taken as a function of the percentage error in the weights. But they assume that the perturbations of input and weight are small and the number of Adalines is large enough. Cheng and Yeung 2 modify the hypersphere model to analyze the sensitivity of Neocognitrons. Unfortunately the hypersphere model they use fails for the case of MLP because its input and weight vectors generally cannot span a whole hypersphere surface.

Piche 3 employs a statistical geometrical argument to analyze the effects of weight errors in Madalines. He makes the assumption that inputs and weights as well as their

errors are all independently identically distributed (i.i.d.) with mean zero. Based on such a stochastic model and under the condition that both input and weight errors are small enough, Piche derives an analytical expression for the sensitivity as the ratio of the variance of the output error to the variance of the output. Piche's method is generalized by Yeung and Sun 4 and they apply the method to the neural networks with antisymmetric squashing activation functions. When using the method, they removed the restriction on the magnitude of input and output errors. Because too strong assumptions are made in the stochastic model, the results gained in this way can only be applied to an ensemble of neural networks, but not to an individual one.

Basically, sensitivity studies on MLPs can be divided into two approaches: the analytical approach and the statistical one. Sensitivity is defined as a partial derivative of output to input in the analytical approach. With this method, Hashem 5 and Fu and Chen 6 compute the partial derivative one layer at a time, starting from the output layer and proceeding backward (backward chaining is used) toward the input layer. It's also used by Zurada et al. 7 and Engelbrecht et al. to delete redundant inputs and prune the architecture of the MLP in 8 and 9. Choi and Choi 10 define the sensitivity as the ratio of the standard deviation of output errors to the standard deviation of the weight or input errors under the condition that the latter tends to zero with the statistical approach. They introduce the statistical sensitivity measure for the MLPs with differentiable activation functions and obtain a formula for the sensitivity of a single-output MLP with fixed weights and a given input pattern. From the above, it's known that these two approaches can only be used to measure the output error with respect to a given input pattern for a specific MLP while they are not applicable in measuring the expected output error with respect to the overall input patterns, which is an important factor considering the performance measurement of neural networks, especially for continuous neural networks like MLPs.

Broomhead and Lowe 11 are the first ones to exploit the use of RBF in the design of the neural networks. Different definitions of the sensitivity of RBFNNs are proposed and applied to many fields. In 13 the sensitivity for the first feature is defined as the variance of the partial derivative of the perturbed output with respect to the first feature. In 14 the sensitivity for the first feature is defined as the mathematical expectation of the square of the partial derivative of the output with respect to the first feature. In 12 and 15 Yeung et al. propose Localized Generalization Error model in which a term is defined as sensitivity and they apply this model to Architecture Selection and Feature Selection of RBFNN. But the relationship between the sensitivity and the parameters of the network is not given. In this paper, we compute and analyze the relationship between the sensitivity and the input errors and the perturbation of others parameters of the RBFNN.

2 RBFNN Model

2.1 Notation

There are three layers with the construction of a RBFNN: input layer, hidden layer and output layer, with entirely different neurons. For a better description of the RBFNN model, we define some notations as follows:

- n : Number of neurons in the input layer (or the dimensionality of the input vector).
- m : Number of the centers (or weights).
- x_i : The value of the i -th neuron in the input layer.
- Δx_i : The error of x_i .
- $x = (x_1, x_2, \dots, x_n)^T$: The input vector of RBFNN.
- $\Delta x = (\Delta x_1, \Delta x_2, \dots, \Delta x_n)^T$: The error x .
- w_j : Weight between output and the j -th center.
- $\mu_j = (\mu_{j1}, \mu_{j2}, \dots, \mu_{jn})^T$: The j -th center.
- $\Delta \mu_j = (\Delta \mu_{j1}, \Delta \mu_{j2}, \dots, \Delta \mu_{jn})^T$: The error of the j -th center μ_j .
- o_j : Output of the j -th neuron of the hidden layer.
- v_j : Width of the j -th center.
- y : The value of the output.
- Δy : The error of y .

2.2 Structure of a RBFNN

Based on the above notation, Figure 1 shows the structure of a RBFNN. The n source neurons in input layer receive the signal from environment; the hidden layer consists of m computation neurons; each hidden neuron is mathematically described by a RBF:

$$o_j(x) = \varphi(x - \mu_j) \tag{1}$$

Thus, unlike a multilayer perceptron, the links connecting the source neurons to the hidden neurons are direct connections with no weights; and the output layer consists of a single computational neuron which applies a simple summation of the weighted responses.

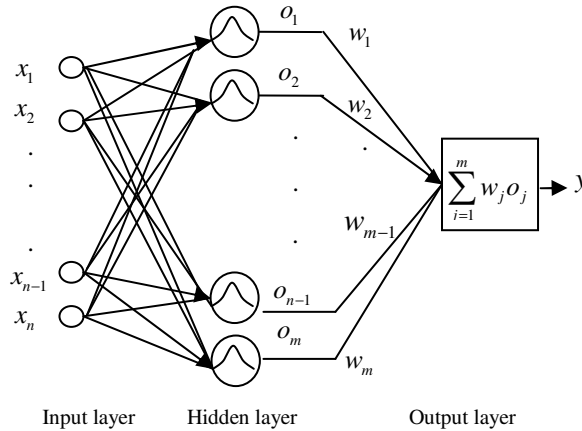


Fig. 1. Structure of a RBFNN

We focus on the use of a Gaussian function as the RBF, in which case each computational neuron in the hidden layer of the network is defined by

$$\begin{aligned}
 o_j(x) &= \varphi(x - \mu_j) \\
 &= \exp\left[-\frac{1}{2v_j^2}\|x - \mu_j\|^2\right] \\
 &= \exp\left[-\frac{1}{2v_j^2}\sum_{i=1}^n(x_i - u_{ji})^2\right], \tag{2}
 \end{aligned}$$

so a RBFNN could be described as:

$$\begin{aligned}
 y = f(x) &= \sum_{j=1}^m w_j o_j(x) \\
 &= \sum_{j=1}^m w_j \exp\left[-\frac{1}{2v_j^2}\sum_{i=1}^n(x_i - u_{ji})^2\right], \tag{3}
 \end{aligned}$$

Then the output error due to the input error can be expressed as:

$$\Delta y = f(x + \Delta x) - f(x). \tag{4}$$

3 Sensitivity Computation of the RBFNN

For computing the sensitivity of a RBFNN, some notations are defined as follows:

$X_i, (i = 1, \dots, n)$: Random variable for the values of the i -th neurons in the input layer

$\Delta X_i, (i = 1, \dots, n)$: Random variable for the error of X_i

$X = (X_1, X_2, \dots, X_n)^T$: Random vector for the values of the input vector

$\Delta X = (\Delta X_1, \Delta X_2, \dots, \Delta X_n)^T$: Random vector for the error of X

$X_p = X + \Delta X$: Random vector for the values of the perturbed input vector.

$W_j, (j = 1, \dots, m)$: Random variable for the values of the weight between the output and the j -th center.

Y : Random variable for the values of the output.

ΔY : The error of Y

$Y_p = Y + \Delta Y$: Random variable for the value of the perturbed output.

The random variable Y can be express as the function of the random vector $X = (X_1, X_2, \dots, X_n)^T$:

$$\begin{aligned}
 Y = f(X) &= \sum_{j=1}^m W_j \exp\left(-\frac{\sum_{i=1}^n (X_i - u_{ji})^2}{2v_j^2}\right) \\
 &= \sum_{j=1}^m W_j \exp(S_j) = \sum_{j=1}^m W_j O_j \tag{5}
 \end{aligned}$$

where
$$S_j = -\frac{1}{2v_j^2} \|X - u_j\|^2 = -\frac{1}{2v_j^2} \sum_{i=1}^n (X_i - u_{ji})^2$$
 and

$O_j = \exp(S_j)$. And in the similar way the random variable Y_p can be express as:

$$Y_p = f(X + \Delta X) \tag{6}$$

As mentioned in 16 and 17 sensitivity is a measure being used to evaluate the degree of the response of a network to the perturbation and imprecision of its input and parameters. The sensitivity of the network is defined as the absolute value of the mathematical expectation of ΔY :

$$S = |E(\Delta Y)| = |E(Y_p) - E(Y)| \tag{7}$$

where $E(\cdot)$ denote the operation of the mathematical expectation.

In order to study the effect of the perturbation of the inputs, it is assumed that every input $X_i, (i=1, \dots, n)$ is i.i.d. and obeys the uniform distribution on interval (a, b) , i.e., $X_i \sim U(a, b), (i=1, \dots, n)$. Then the value of S can be evaluated by the Central Limit Theorem (CLT) and the steps are as follows.

According to the CLT, when X_1, X_2, \dots, X_n are independent and identically distributed and the numbers of the neurons in the input layer are not too few, S_j would have Normal distributions:

$$S_j \sim N(E(S_j), D(S_j))$$

where $D(\cdot)$ denote the operation of the variance. Because $X_i, (i=1, \dots, n)$ are i.i.d., the value of $E(S_j)$ and $D(S_j)$ can be computed as:

$$\begin{aligned} E(S_j) &= -\frac{1}{2v_j^2} \sum_{i=1}^n E(X_i - u_{ji})^2 \\ &= -\frac{1}{2v_j^2} \sum_{i=1}^n [DX_i + (EX_i - u_{ji})^2] \end{aligned} \tag{8}$$

$$\begin{aligned} D(S_j) &= \frac{1}{4v_j^4} \sum_{i=1}^n D(X_i - u_{ji})^2 \\ &= \frac{1}{4v_j^4} \sum_{i=1}^n \left[EX_i^4 - (EX_i^2)^2 + 4u_{ji}^2 DX_i \right. \\ &\quad \left. - 4u_{ji} EX_i^3 + 4u_{ji} EX_i^2 EX_i \right] \end{aligned} \tag{9}$$

Then by the knowledge of probability theory we can conclude that $O_j = \exp(S_j)$ would have Log-Normal distributions and the mathematical expectation of O_j can be expressed by $E(S_j)$ and $D(S_j)$ as follows:

$$E(O_j) = E[\exp(S_j)] = \exp\left[E(S_j) + \frac{(DS_j)^2}{2}\right] \tag{10}$$

Thus the mathematical expectation of O_j can be obtained by substituting formula (8) and (9) into formula (10).

So the mathematical expectation of Y can be obtained by substituting formula (10) into the follow formula:

$$E(Y) = E[f(X)] = E\left(\sum_{j=1}^m W_j O_j\right) = \sum_{j=1}^m W_j E(O_j) \tag{11}$$

In order to compute the value of $E(Y_p)$, the error ratio of the input X_i , denoted by δX_i , is defined to be the ratio of ΔX_i to X_i :

$$\delta X_i = \frac{\Delta X_i}{X_i}, (i=1, \dots, n). \tag{12}$$

But if for different i the value of δX_i is different, it is complicated to compute the sensitivity S and that is why we let δX_i be the same value δ for any i :

$$\delta X_i = \delta, (i=1, \dots, n).$$

Then the value of $E(Y_p)$ in formula (7) can be expressed as:

$$E(Y_p) = E[f(X + \Delta X)] = E\{f[(1 + \delta)X]\} \tag{13}$$

Under the aforementioned assumption that the variable $X_i, (i=1, \dots, n)$ obeys the uniform distribution on interval (a, b) , we have:

when $1 + \delta > 0$,

$$(1 + \delta)X_i \sim U((1 + \delta)a, (1 + \delta)b);$$

when $1 + \delta < 0$,

$$(1 + \delta)X_i \sim U((1 + \delta)b, (1 + \delta)a).$$

It is obviously that $1 + \delta$ is usually positive, i.e., $1 + \delta > 0$, so we can obtain $E(Y_p)$ by substituting $(1 + \delta)a$ for a , $(1 + \delta)b$ for b in formula (11). After obtaining the unperturbed result $E(Y)$ and the perturbed result $E(Y_p)$, the sensitivity can be computed easily by substituting $E(Y)$ and $E(Y_p)$ to formula (7).

4 Sensitivity Analysis of the RBFNN

To study the relationship between the sensitivity of the RBFNN and its input, a number of simulations are conducted. It is found that the theoretical results coincide with the experimental ones, which confirm the feasibility of the expressions. All inputs in the

network are assumed to be i.i.d. with uniform distribution on interval (a,b) . In the simulations we set $a = 0, b = 1$, i.e. we can substitute $a = 0, b = 1$ into formula (11) to compute $E(Y)$ and substitute $a = 0, b = 1 + \delta$ into formula (11) to compute $E(Y_p)$, then compute the sensitivity S by formula (7). The experimental results and the theoretical ones are obtained by taking an average of 1000 individual computer runs.

From the results shown in Figure 2-5, some results on the sensitivity of RBFNN are stated in the following. In the following figures, the dash and the solid lines are respectively the experimental and the theoretical results, and the vertical axis represents the sensitivity of RBFNNs.

4.1 Relationship Among the Sensitivity, the Input Error Ratios and the Number of Input

4.1.1 Input Error Ratios

Figure 2 shows the relationship between the sensitivity and the input error ratios. The horizontal axis represents the input error ratios. From Figure 2, it can be concluded that:

- (1). The sensitivity increases with its input error ratios increasing for any n .
- (2). When n is small, the difference between the theoretical and experimental results is big and when n is large, the theoretical and experimental result are almost equal. The key reason is that the distribution of S_j will tend to be normal distribution as the number of the input increases according to the CLT.
- (3). When n is about 8, the increase of the sensitivity is the fastest and the increases is slowly when n is large.

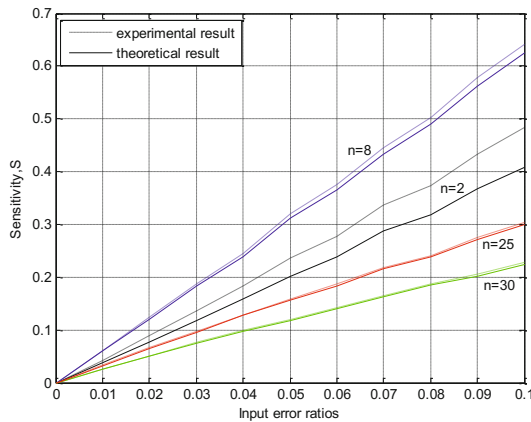


Fig. 2. Sensitivity to input error for different n

4.1.2 Number of the Neurons in the Input Layer

The relationship between the sensitivity and the number of the neurons in the input layer for the different input error ratios is shown in Figure 3. In this figures, the

horizontal axis represents the number of neurons in the input layer. When giving a perturbation to the input, it is obvious:

- (1). When the error ratios are small, the sensitivity may first moderately increase with small n and then moderately decreases with n increasing;
- (2). When the error ratios are large, the sensitivity may first increase with small n and then decreases with n increasing.
- (3). For different input error ratios, the sensitivity reaches the maximum when $n = 7$ or $n = 8$.

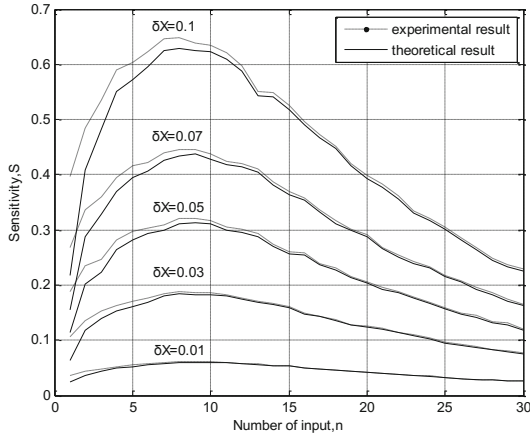


Fig. 3. Sensitivity to the number of the input neurons for different δ

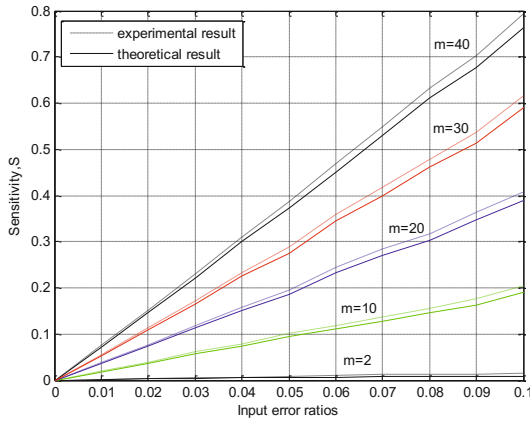


Fig. 4. Sensitivity to input error for different m

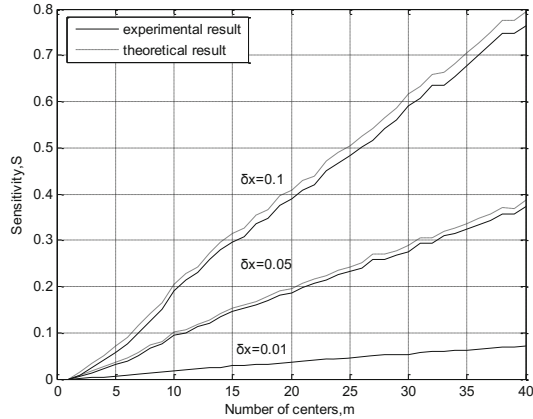


Fig. 5. Sensitivity to the number of centers for different δ

4.2 Relationship Among the Sensitivity, the Input Error Ratios and the Number of Centers

Figure 4 and 5 show the relationship among the sensitivity, the input error ratios and the number of centers. Here we set the number of the input neurons to 6.

4.2.1 Input Error Ratios

Figure 4 shows the relationship between the sensitivity and the input error ratios for different m . The horizontal axis represents the input error ratios. From Figure 4, it can be concluded that:

- (1). The sensitivity increases with its input error ratios increasing for any m .
- (2). When m is small, the sensitivity gently increases while it increases rapidly when m is large.

4.2.2 Number of the Centers, m (or the Number of the Neurons in the Hidden Layer)

Figure 5 shows the following relationship between the sensitivity and the number of the centers. In these figures, the horizontal axis represents the number of the centers.

- (1) For different δ , the sensitivity increases with the number of the centers.
- (2) When the error ratios are small, the sensitivity increases slowly while it increases rapidly when the error ratios are large.

5 Conclusions

This paper proposes a method to compute the sensitivity of RBFNN due to the input errors based on CLT. When the number of the neurons in the input layer is not too few, the theoretical results coincide with the experimental ones, which verify the reliability

and feasibility of the method. It can be concluded that both for different n and for different m the sensitivity increase with the increasing of the input error ratios; the sensitivity increases with the number of the neurons in the hidden layer (or the number of the centers) for any input error ratios. But the sensitivity first increases and then decreases with the increasing of n .

Acknowledgements. This work is supported by Baoding Science and Technology Bureau (12ZG030), Hebei University Foundation (2010Q27), Hebei Natural Science Foundation (A2011201053) and Hebei Provincial Education Department (2010110). The authors also thank the reviewers for careful reading the manuscript and for many helpful comments.

References

1. Stevenson, M., Winter, R., Widrow, B.: Sensitivity of feedforward neural networks to weight errors. *IEEE Trans. Neural Networks* **1**, 71–80 (1990)
2. Cheng, A.Y., Yeung, D.S.: Sensitivity analysis of neocognitron. *IEEE Trans. Syst., Man, Cybern. C* **29**, 238–249 (1999)
3. Piche, S.W.: The selection of weight accuracies for Madalines. *IEEE Trans. Neural Networks* **6**, 432–445 (1995)
4. Yeung, D.S., Sun, X.: Using Function Approximation to Analyze the Sensitivity of MLP with Antisymmetric Squashing Activation Function. *IEEE Transactions on Neural Networks* **13**(1), 34–44 (2002)
5. Hashem, S.: Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. In: *Proc. IJCNN 1992 Baltimore, MD*, vol. 1, pp. 419–424 (1992)
6. Fu, L., Chen, T.: Sensitivity analysis for input vector in multilayer feedforward neural networks. In: *Proc. IEEE Int. Conf. Neural Networks San Francisco, CA*, vol. 1, pp. 215–218 (1993)
7. Zurada, J.M., Malinowski, A., Usui, S.: Perturbation method for deleting redundant inputs of perceptron networks. *Neurocomput.* **14**, 177–193 (1997)
8. Engelbrecht, A.P., Cloete, I.: A sensitivity analysis algorithm for pruning feedforward neural networks. In: *Proc. IEEE Int. Conf. Neural Networks, Washington, DC*, vol. 2, pp. 1274–1277 (1996)
9. Engelbrecht, A.P., Fletcher, L., Cloete, I.: Variance analysis of sensitivity information for pruning feedforward neural networks. In: *Proc. IEEE Int. Conf. Neural Networks, Washington, DC*, pp. 1829–1833 (1999)
10. Choi, J.Y., Choi, C.-H.: Sensitivity analysis of multilayer perceptron with differentiable activation functions. *IEEE Trans. Neural Networks* **3**, 101–107 (1992)
11. Broomhead, D.S., Lowe, D.: Multivariate functional interpolation and adaptive networks. *Complex Systems* **2**, 321–355 (1988)
12. Yeung, D.S., Ng, W.W.Y., Wang, D., Tsang, E.C.C., Wang, X.-Z.: Localized Generalization Error and Its Application to Architecture Selection for Radial Basis Function Neural Network. *IEEE Trans. on Neural Networks* **18**(5), 1294–1305 (2007)
13. Wang, X.-Z., Li, C.: A New Definition of Sensitivity for RBFNN and Its Applications to Feature Reduction. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) *ISNN 2005. LNCS*, vol. 3496, pp. 81–86. Springer, Heidelberg (2005)

14. Wang, X.Z., Zhang, H.: An Upper Bound of Input Perturbation for RBFNNs Sensitivity Analysis. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 4074–4079 (August 18-21, 2005)
15. Ng, W.W.Y., Yeung, D.S., Firth, M., Tsang, E.C.C., Wang, X.-Z.: Feature Selection Using Localized Generalization Error for Supervised Classification Problems for RBFNN. *Pattern Recognition*, 3706–3719 (2008)
16. Zeng, X., Yeung, D.S.: Sensitivity analysis of multilayer perceptron to input and weight perturbations. *IEEE Trans. Neural Netw.* **12**(6), 1358–1366 (2001)
17. Yang, S., Ho, C., Siu, S.: Computing and Analyzing the Sensitivity of MLP Due to the Errors of the i.i.d. Inputs and Weights Based on CLT. *IEEE Trans. on Neural Networks* **21**(12) (2010)

Fuzzy If-Then Rules Classifier on Ensemble Data

Tien Thanh Nguyen^{1(✉)}, Alan Wee-Chung Liew¹, Cuong To¹,
Xuan Cuong Pham², and Mai Phuong Nguyen³

¹ School of Information and Communication Technology,
Griffith University, Queensland, Australia
tienthanh.nguyen2@griffithuni.edu.au,
{a.liew,c.chieuto}@griffith.edu.au

² School of Information and Communication Technology,
Hanoi University of Science and Technology, Hanoi, Vietnam
cuong.phamxuan.hust@gmail.com

³ College of Business Massey University, Massey, New Zealand
phuongnm0590@gmail.com

Abstract. This paper introduces a novel framework that uses fuzzy IF-THEN rules in an ensemble system. Our model tackles several drawbacks. First, IF-THEN rules approaches have problems with high dimensional data since computational cost is exponential. In our framework, rules are operated on outputs of base classifiers which frequently have lower dimensionality than the original data. Moreover, outputs of base classifiers are scaled within the range [0, 1] so it is convenient to apply fuzzy rules directly instead of requiring data transformation and normalization before generating fuzzy rules. The performance of this model was evaluated through experiments on 6 commonly used datasets from UCI Machine Learning Repository and compared with several state-of-art combining classifiers algorithms and fuzzy IF-THEN rules approaches. The results show that our framework can improve the classification accuracy.

Keywords: Ensemble method · Combining multiple classifiers · Classifier fusion · Fuzzy IF-THEN rules

1 Introduction

Ensemble classification is a class of methods which combine models to achieve lower error rate than using a single model. The concept of “model” in ensemble methods includes not only using different learning algorithms, or using different training set for the same learning algorithm, but also generating generic classifiers in combination to improve accuracy of classification task.

In recent years, ensemble methods have been studied extensively [20]. Choosing a suitable algorithm for a given dataset is a non-trivial problem and is usually based on trial and error or experience. Two factors that need to be considered are classification accuracy and execution time. In many cases, accuracy in classification is preferred. However, in online learning, the time factor also plays an important role. In this situation, choosing a learning algorithm that exhibits fast execution with acceptable accuracy level could be the primary consideration. Ensemble methods can achieve higher accuracy than a single learning algorithm while using simple classifiers.

© Springer-Verlag Berlin Heidelberg 2014

X. Wang et al. (Eds.): ICMLC 2014, CCIS 481, pp. 362–370, 2014.

DOI: 10.1007/978-3-662-45652-1_36

Fuzzy rule based classifier that extracts knowledge from data has been successfully applied to many classification tasks [21]. In general, a fuzzy IF-THEN rule R corresponding with class k is given by:

$$\begin{aligned} \text{Rule } R: & \text{ If } X_{i_1} \in L_1 \text{ and } X_{i_2} \in L_2 \text{ and...and } X_{i_D} \in L_D \\ & \text{ then } X_i \in \text{ class } k \text{ with } CF \end{aligned} \quad (1)$$

in which $X_i = (X_{i_1}, \dots, X_{i_D})$ is a D-vector, L_i is an antecedent fuzzy set and CF is called rule weight (i.e., certainty factor) in the range $[0, 1]$.

A problem with fuzzy IF-THEN rules is that it is impractical with high dimensional data since if we have a D-dimension dataset and $|L|$ antecedent fuzzy set, the total number of rules in rule set is $|L|^D$. Hence, the number of rules increases exponentially with the dimensionality of the feature space. Moreover, the components in the feature vector need to be transformed and normalized before rule generation or specific antecedent fuzzy set needs to be built for each of them. Our approach overcomes these problems.

The paper is organized as follows. Section 2 provides a review on several state-of-art combining classifiers algorithms and fuzzy rules based classification. In Section 3, we present our fuzzy rule based ensemble classification framework. Experimental results on 6 popular UCI datasets are presented in Section 4. Finally, we summarize our work and propose several future extensions.

2 Related Work

There are many combining algorithms proposed in which algorithms based on stacking are the most popular [2]. In this model, the training set is divided into B nearly equal non-overlapping parts. One part plays the role of test set in turn and the rest as training set so all observations are tested once. The outputs of stacking are posterior probabilities that observations belong to a class according to each classifier. The set of posterior probabilities of all observations is called meta-data or Level1 data to distinguish it from Level0 data which is the original data.

To apply stacking to ensemble of classifiers, Ting and Witten [2] proposed Multiple Response Linear Regression algorithm (MLR) to combine posterior probabilities from Level1 data of each observation based on sum of weights calculated from K linear regression functions. Kuncheva et al. [1] applied fuzzy relation to find the relationship between posterior probability and Decision Template for each class computed on posterior probability of observations and its true class label. They also proposed eleven measurements between two fuzzy relations to predict class label of observations. Merz [4] combined Stacking, Correspondence Analysis and K Nearest Neighbor (KNN) in a single algorithm called SCANN. The idea is to use Correspondence Analysis to analyze relationship between rows (observations) and columns (outputs of Stacking and true label of each observation) to form a new representation for outputs from base classifiers. After that, KNN is applied to that representation to predict the label of an observation.

On the other hand, popular and simple combining classifiers algorithms was introduced by Kittler et al. [3], where they presented six fixed combining rules, namely Sum, Product, Vote, Min, Max and Average for combination. A benefit of fixed rules is that no training on Level1 data is required.

The first idea about fuzzy IF-THEN rules in classification system was introduced by Valenzuela and Rendon [6] and further developed [7-19]. Recently, several approaches have been introduced to reduce the number of generated rules while maintaining classification accuracy. Verikas et al. [14] limit classifier structure by using SOM tree algorithm and then optimize the number of input variable for each rule by applying Genetic Algorithm (GA). By using that approach, several input variables were eliminated from the model so the number of generated rules is reduced. Soua et al. [17] introduced a rule reduction strategy by computing the correlation between each pair of component in feature vector, and then gathering high-correlation components in a single group, finally building fuzzy rules for each group. However, these approaches still produce high error rate in the experiment.

3 Proposed Framework

The most important distinction between our work and previous works is that we apply fuzzy rules based classification on Level1 data. Attributes in Level0 data are frequently different in nature, measurement unit, and type. Level1 data, on the other hand, can be viewed as scaled result from feature domain to posterior domain where data is reshaped to be all real values in [0, 1]. Observations that belong to the same class will have nearly equal posterior probabilities generated from base classifiers and would locate close together in the new domain. Moreover, as Level1 data has lower dimension than Level0 data, fuzzy rules-based classifier on Level1 data would be higher performance since number of rules is generated in a small value.

Figure 1 shows an overall view at our framework. We first apply Stacking on the training set to generate the outputs of base classifiers. After that, fuzzy IF-THEN classification system is built on Level1 data by a 4-step process:

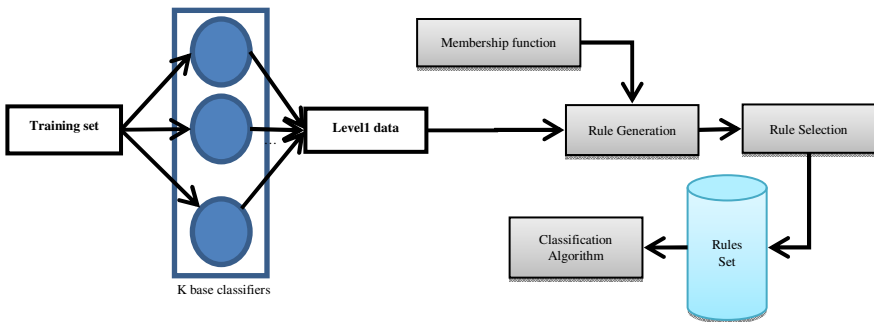


Fig. 1. Proposed framework

Step1 (Construction of membership functions): This step is also called *fuzzy discretization* in which membership function is introduced for each component of feature vector. Common membership functions are triangular or trapezoidal. In our framework, we cover the universe of discourse using only five fixed membership functions (Figure 2) and a special “don’t care” function membership [10] (with the membership function assuming 1 for all values of a variable).

Step2 (Rule Generation and Reduction): Based on the constructed membership functions, fuzzy rules are generated for each attributes of feature vector. Rules with small or zero weight are deleted from the rule set. In our framework, we limit the number of fuzzy attributes to three to reduce the number of generated rules.

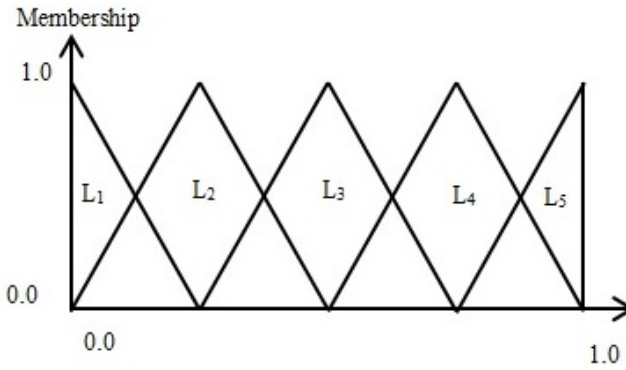


Fig. 2. Triangular membership functions

Step3 (Rule Selection): The goal of this task is to reduce the number of rules in the rule set so as to save computation cost and boost performance. In this paper, we select rules from rule set based on both confidence and support of rule defined by:

$$c(R_j \rightarrow \text{class } k) := \frac{\sum_{X \in \text{class } k} \mu_{R_j}(X)}{\sum_{X \in \text{Training Set}} \mu_{R_j}(X)} \tag{2}$$

$$s(R_j \rightarrow \text{class } k) := \frac{\sum_{X \in \text{class } k} \mu_{R_j}(X)}{N} \tag{3}$$

where N is the number of observations in the training set and

$$\mu_{R_j}(X) = \prod_{i=1}^D \mu_{R_j}(X_i) \tag{4}$$

is the degree of observation X belonged to Rule R_j computed by multiplying the degrees of all D components X_i of X ($i = \overline{1, D}$) that refer to rule R_j

Rules from the same class are grouped together and their confidence (2) and support (3) are multiplied together and then sorted in descending order as rank of a rule. T rules with highest rank's value for each class and save them in a rule set database (denoted by S).

Step4 (Classification based on Rule set): An observation is predicted to belong to a certain class by using the fuzzy rules in S . For example, in single winner strategy, an unlabeled observation $XTest$ is predicted to be of class k^* by:

$$XTest \in \text{class } k^* \text{ if } k^* = \arg \max_j (\mu_{R_j}(XTest) \times CF_j) \tag{5}$$

For voting strategy, the classification decision is given by:

$$XTest \in \text{class } k^* \text{ if } k^* = \arg \max_k (\sum_{R_j \text{ corresponding with class } k} \mu_{R_j}(XTest) \times CF_j) \tag{6}$$

where CF_j is rule weight associated with rule $R_j \in S$

In this paper, single winner strategy (5) is selected as the decision scheme. The algorithm is presented below.

Algorithm: Ensemble system with Fuzzy IF-THEN Rules

Training process (Input: Training set, base classifiers, number of retained rules for each class: T, Output: Rule set)

Step 1: Compute Level1 data of training set.

Step 2: Define membership function

Step 3: Generate rules based on membership function on Level1 data. Reduce rules with weight=0

Step 4: Compute the product (confidence x support) as rank of each rule, sort them in descending order.

Step 5: Retain T rules related to each class with highest rank, save in a rule set

Test process (Input: unlabeled observation XTest, Output: predicted label of XTest)

Step1: Generate Level1 data of XTest

Step2: For each rule in rule set

Compute membership degree of XTest (4)

End

Predict label of XTest based on (5)

4 Experimental Results

We conducted experiments on six datasets downloaded from UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets.html>). Information of these datasets is listed in Table 1. We kept maximal 40 rules for each class as arbitrary value. Actually, that number should be changed and reported the corresponding error rate [8]. Here, because of space limitation, we only chose a fixed value. We employed three traditional classifiers, namely Linear Discriminant Analysis (LDA), Naive Bayes, K Nearest Neighbor (with K set to 5), to be the base classifiers. Since these classifiers

Table 1. UCI datasets in our experiment

File Name	# of Attributes	# of Classes	# of observations
Bupa	6	2	345
Fertility	9	2	100
Sonar	60	2	208
Iris	4	3	150
Artificial	10	2	700
Pima	6	2	768

have different approaches, diversity of meta-data is ensured. As we only had a single dataset, 10-fold cross validation (CV) technique is used. To ensure objectivity of evaluation, we ran 10-fold CV 10 times to obtain 100 test results.

Table 2. Classification error rate on UCI datasets by fixed combining rules

File Name	Sum		Product		Max		Min		Median		Vote	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3028	4.26E-03	0.3021	4.12E-03	0.2986	4.15E-03	0.2970	4.89E-03	0.3428	4.46E-03	0.3429	4.04E-03
Fertility	0.129	2.46E-03	0.129	2.26E-03	0.127	1.97E-03	0.128	2.02E-03	0.133	2.81E-03	0.131	2.34E-03
Sonar	0.2259	9.55E-03	0.2285	9.81E-03	0.2260	7.01E-03	0.2298	9.32E-03	0.2104	1.00E-02	0.2079	8.16E-03
Iris	0.0387	2.59E-03	0.0407	2.39E-03	0.0440	3.13E-03	0.0413	2.56E-03	0.0333	1.64E-03	0.0327	1.73E-03
Artificial	0.2230	2.06E-03	0.2193	2.05E-03	0.2450	2.57E-03	0.2453	2.90E-03	0.3089	1.36E-03	0.3073	1.03E-03
Pima	0.2405	1.62E-03	0.2419	1.63E-03	0.2411	1.69E-03	0.2449	2.02E-03	0.2376	1.69E-03	0.2365	2.10E-03

Table 3. Classification error rate on UCI datasets by other combining algorithms

File Name	Decision Template		MLR		SCANN		Select best from fixed combining rules		Proposed Method	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3348	7.10E-03	0.3033	4.70E-03	0.3304	4.29E-03	0.2986	4.15E-03	0.3391	6.02E-03
Fertility	0.4520	3.41E-02	0.1250	2.28E-03	x	x	0.127	1.97E-03	0.1100	4.90E-03
Sonar	0.2129	8.80E-03	0.1974	7.20E-03	0.2128	8.01E-03	0.2079	8.16E-03	0.2067	7.50E-03
Iris	0.040	2.50E-03	0.0220	1.87E-03	0.0320	2.00E-03	0.0327	1.73E-03	0.0330	2.89E-03
Artificial	0.2433	1.60E-03	0.2426	2.20E-03	0.2374	2.12E-03	0.2193	2.05E-03	0.2257	2.02E-03
Pima	0.2482	2.00E-03	0.2432	2.30E-03	0.2384	2.06E-03	0.2365	2.10E-03	0.2460	2.89E-03

For comparison, we ran six fixed combining rules because it is simple and popular in combining classifiers systems (Table 2). We also evaluated several state-of-art combining algorithms, namely Decision Template, MLR, and SCANN (Table 3). We noted that SCANN cannot be run on Fertility dataset because its indicator matrix is singular [4]. Besides, results of twelve fuzzy rules approaches published previously are reported in Table 4. Since our model used fuzzy rules-based approach to form the

classification framework so it is important to have a comparison with other fuzzy rules-based classification methods [8, 11, 13-19]. Also our model is a combining classifiers model so it is necessary to compare it with other well-known combining classifiers algorithms [1-4].

Table 4. Classification error rate on UCI datasets by several fuzzy rules approaches

	Ishibuchi et al. method [8, 11]	Soua et al. method [17] SIFCO	Soua et al. method [17] SIFCO-PAF	Wang et al. method [15] refinement	Wang et al. method [15] weight refinement	Verikas et al. method [14]
Bupa	x	x	x	x	x	x
Sonar	0.2406	0.2837	0.3029	0.2	0.29	x
Iris	0.02	0.033	0.0267	0.06	0.07	x
Pima	x	x	x	0.25	0.31	0.2677
	Kim et al. method [19]	Gonzalez et al. method [16] SLAVE 1	Gonzalez et al. method [16] SLAVE 2	Mansoori et al. method [13] SGERD	Angelov et al. method [18] eClass1(offline)	Angelov et al. method [18] eClass0(offline)
Bupa	0.2931	x	x	x	x	x
Sonar	0.2248	x	x	0.2480	0.2343	0.2924
Iris	0.0201	0.043	0.043	0.036	x	x
Pima	0.2310	x	x	0.2692	0.233	0.307

Our framework is equally competitive with selected best results from fixed combining rules (1 win and 1 loss). It is slightly better than Decision Template (2 wins and 0 loss), MLR (2 wins and 1 loss) and SCANN (1 win and 0 loss). These results were obtained based on paired t-test with α is set to 0.05. Moreover, our model is better than the methods in [8] on Sonar (20.67% vs. 24.06%); [17] on Sonar (20.67% vs. 28.37% and 30.29%); [15] on Sonar (20.67% vs. 29%), Iris (3.3% vs. 7%) and Pima (24.6% vs. 31%); [14] on Pima (24.6% vs. 26.77%), [16] on Iris (3.3% vs. 4.3%); [13] on both Pima (24.6% vs. 26.92%) and Sonar (20.67% vs. 24.8%), eClass0 of [18] on Sonar (20.67% vs. 29.24%) and Pima (24.6% vs. 30.7%); eClass1 of [18] on Sonar (20.67% vs. 23.43%). These results demonstrate the superiority of our model.

There are several cases that our model is worse than other approaches, for instance, [19] on Bupa (33.91% vs. 29.31%), Iris (3.3% vs. 2.01%) and Pima (24.6% vs. 23.1%); eClass1 of [18] on Pima (24.6% vs. 23.3%) and [11] on Iris (3.3% vs. 2%). However, the difference in classification error is quite small in most cases. It should be noted that [11] and [19] used GA to optimize the performance of fuzzy rules system whereas we only use traditional approach for fuzzy rule with limitation placed on the number of attributes and membership function to generate fuzzy rules.

5 Conclusion and Future Development

In this paper, we have introduced a new framework by integrating fuzzy IF-THEN rules with ensemble system. Experimental results on 6 popular UCI datasets demonstrate superior performance of our framework compared with state-of-art combining

classifiers algorithms as well as several fuzzy rule approaches. Our model is highly comprehensible since practitioners can readily understand the mechanism of classification using the rules. Another advantage of our model is its applicability to high dimension data. As the dimensionality of Level1 data is equal to the product of the number of classifiers and the number of classes, for system with few classifiers, Level1 data usually is normalized in $[0,1]$ and has lower dimensions. As a result, it is considerably less complex to apply fuzzy rules to Level1 data than to Level0 data.

In the future, we will investigate an adaptive mechanism for membership functions that are more suitable for meta-data. The relationship between the number of retained rules and the accuracy of classification task will be reported and analyzed. We will also consider applying classifier and feature selection to our model to improve its performance.

References

1. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision Templates for Multi Classifier Fusion: An Experimental Comparison. *Pattern Recognition* **34**(2), 299–314 (2001)
2. Ting, K.M., Witten, I.H.: Issues in Stacked Generation. *Journal of Artificial Intelligence Research* **10**, 271–289 (1999)
3. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3) (March 1998)
4. Merz, C.: Using Correspondence Analysis to Combine Classifiers. *Machine Learning* **36**, 33–58 (1999)
5. Sen, M.U., Erdogan, H.: Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters* **34**, 265–274 (2013)
6. Valenzuela, M., Rendon: The fuzzy classifier system a classifier system for continuously varying variables. In: *Proceeding 4th ICGA*, pp. 346–353 (1991)
7. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems* **141**, 59–88 (2004)
8. Ishibuchi, H., Yamamoto, T.: Comparison of Heuristic Criteria for Fuzzy Rule Selection in Classification Problems. *Fuzzy Optimization and Decision Making* **3**, 119–139 (2004)
9. Ishibuchi, H., Nakashima, T., Morisawa, T.: Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems* **103**, 223–238 (1999)
10. Ishibuchi, H., Murata, T., Turksen, I.B.: Single-objective and two-objective genetic algorithms for selecting linguistic for pattern classification problems. *Fuzzy Sets and Systems* **89**, 135–150 (1997)
11. Ishibuchi, H., Nakashima, T., Murata, T.: Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences* **136**, 109–133 (2001)
12. Murata, T., Ishibuchi, H., Nakashima, T., Gen, M.: Fuzzy partition and input selection by genetic algorithms for designing fuzzy rule-based classification systems. In: *7th International Conference, EP98 San Diego, California, USA* (March 25–27, 1998)
13. Mansoori, E.G., Zolghadri, M.J., Katebi, S.D.: SGERD: A Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data. *IEEE Transactions on Fuzzy Systems* **16**(4) (August 2008)

14. Verikas, A., Guzaitis, J., Gelzinis, A., Bacauskiene, M.: A general framework for designing a fuzzy rule-based classifier. *Knowledge and Information Systems* **29**(1), 203–221 (2011)
15. Wang, X.-Z., Dong, C.-R.: Improving Generalization of Fuzzy IF-THEN Rules by Maximizing Fuzzy Entropy. *IEEE Transactions on Fuzzy Systems* **17**(3) (June 2009)
16. Gonzalez, A., Pérez, R.: SLAVE: A Genetic Learning System Based on an Iterative Approach. *IEEE Transactions on Fuzzy Systems* **7**(2), 176–191 (1999)
17. Soua, B., Borgi, A., Tagina, M.: An ensemble method for fuzzy rule-based classification systems. *Knowledge and Information Systems* **36**(2), 385–410 (2013)
18. Angelov, P.P., Zhou, X.: Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems* **16**(6) (December 2008)
19. Kim, M.W., Ryu, J.W.: Optimized Fuzzy Classification Using Genetic Algorithm. In: *The Second International Conference on Fuzzy Systems and Knowledge Discovery, China* (August 2005)
20. Rokach, L.: Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Journal of Computational Statistics & Data Analysis* **53**(12), 4046–4072 (2009)
21. Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems* **141**(1), 5–31 (2004)

Image Segmentation Based on Graph-Cut Models and Probabilistic Graphical Models: A Comparative Study

Maedeh Beheshti^(✉) and Alan Wee-Chung Liew

School of Information and Communication Technology,
Gold Coast Campus, Griffith University, Queensland 4222, Australia
maedeh.beheshti@griffithuni.edu.au, a.liew@griffith.edu.au

Abstract. Image segmentation has been one of the most important unsolved problems in computer vision for many years. Recently, there have been great effort in producing better segmentation algorithms. The purpose of this paper is to introduce two proposed graph based segmentation methods, namely, graph-cut models (deterministic) and a unified graphical model (probabilistic). We present some foreground/background segmentation results to illustrate the performance of the algorithms on images with complex background scene.

Keywords: Image segmentation · Graph-cut segmentation · Graphical models

1 Introduction

Partitioning an image into meaningful parts or regions that share certain visual characteristics based on image attributes like level of brightness, colour, texture, depth or motion is the main purpose of image segmentation. Image segmentation plays a crucial role in many applications such as object identification and recognition, content-based image retrieval, and medical image analysis. In spite of the many existing algorithms and techniques for image segmentation, in this paper we focus only on graph partitioning methods. Graph based segmentation, which is based on different principles of many traditional segmentation algorithms, is a relatively new segmentation method. Graph based segmentation algorithms can be broadly divided into two types 1- deterministic 2- probabilistic [1]. Algorithms such as max/min cut and normalized cut belong to the deterministic approach, and algorithms such as Markov Random Field and Bayesian Network belong to the probabilistic graphical approach. Energy minimization and inference are the main tasks in graph cut algorithms and probabilistic models, respectively. Because of dealing with many variables, the energy function of graph-cut algorithms tends to be very complex and often NP-hard, especially for non-convex problems [2][3][4].

In this paper, we describe two representative graph-based segmentation algorithms, coming from each type. We introduce readers to the idea behind each method, and present some experimental evaluation to demonstrate their empirical performance.

2 Graph-Cut Segmentation Algorithms

2.1 Binary and Multi-Label Graph-Cut

Boykov and Jolly [2] proposed a graph-cut method for segmentation, which involves finding the s-t cut of minimal total cost with 2 labels (binary labels) in a graph. In their approach, an image is to be segmented into object and background, with soft constraints on the boundary and region. In their model, an undirected graph $G(V, E)$ (see Figure 1) is formed in which V and E represent graph nodes (equal to image pixels P) and graph edges, respectively. In addition, two extra terminal nodes, S (source) and T (sink), which represent the object and background labels, are also defined. All edges in this graph are divided into two classes: n-links and t-links, where n-links are edges between pixels and t-links are edges between pixels and terminal nodes.

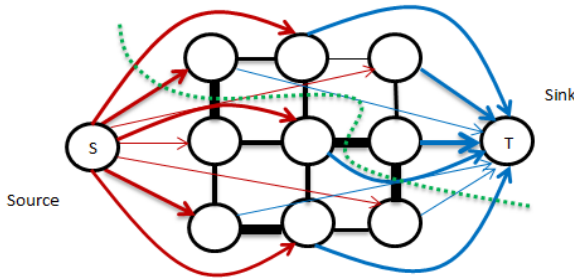


Fig. 1. A graph with two terminals – Arrows show the flow from source (S) to sink (T) – dashed curve represents the min-cut point [2][3].–Thickness of the links represent the amount of flows that has been sent from the source to the destination

Each non-terminal node has two terminal links, one for S and one for T , and a number of n-links. The n-links or neighborhood links are the links between pixels and they construct an arbitrary neighborhood system. Each edge has associated some cost or weight. An optimal s-t cut is a subset of edges with minimum cost which partition the graph (image) into two parts, source and terminal. In this regard, the purpose of segmentation is to minimize the energy function in(1) which is the sum of regional (cost of t-links) and boundary terms (cost of n-links).

$$E(A) = \delta \cdot Region(A) + Boundary(A) \tag{1}$$

In (1), A defines a segmentation and $\delta \geq 0$ is a coefficient which emphasizes the region term. The segmentation is done based on the max flow algorithm [5] which we will elaborate on in section 2.4.

To handle more than two labels, i.e. $|L| \geq 3$ considering multi-label energy minimization through multi-terminal min-cut or move-making (local search) algorithms like alpha-beta swap algorithm and alpha expansion was proposed [4]. Alpha-beta swap algorithm and alpha expansion are two kinds of very large search neighborhood techniques (VLSN). Through VLSN algorithms and appropriate move

space, an exploration of an exponential number of alternative labeling will be facilitated in polynomial time. Due to the lack of space we only consider $\alpha\beta$ -swap here. Given current labeling f from a pair of labels $\{\alpha, \beta\} \in L$ the purpose is finding a labeling $f' \in \{\alpha, \beta\}$ that minimizes the energy function in equation (2) [4]. According to Figure 2 each variable labeled either α or β is able to keep its current label (See Figure 2.a) or swap to the other labels. Figure 2.b represents the swap between α, β labels, and variables with label β have been expanded and variables with label α have been decreased. In addition, Figure 2.c represents the swap between β, γ labels where variables with label β have been decreased and variables with label γ have been increased. Each time only two variables from move space will be selected for swapping. If there are $K1$ variables with a current label in $\{\alpha, \beta\}$ there will be 2^{K1} possible $\alpha\beta$ -swap moves available. Consider the following energy function

$$Energy(f) = Energy_{smooth}(f) [\sum_{(p,q) \in N} V_{p,q}(f_p, f_q)] + Energy_{data}(f) [\sum_{p \in P} D_p(f_p)] \tag{2}$$

where N is the set of interacting pairs of pixels such as p and q , $D_p(l_p)$ is the cost of assigning a label such as l to a pixel p , and $V_{p,q}(f_p, f_q)$ is a smooth term between two pixels p and q that shows coherence and consistency between labels [3][4]. The first part of (2) expresses an extent to which f is not piecewise smooth while the second part measures the disagreement between f and the observed data [4]. The smooth term can be viewed as a regularization term as it reduces inconsistency and fluctuations caused by noisy data.

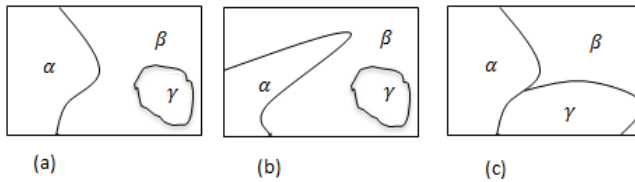


Fig. 2. The left figure shows current labeling containing three α, β and γ parts - The middle and right figures show an α, β and γ, β swap respectively

2.2 Optimal Multi-Region Object Segmentation

In the optimal multi-region model of [6], one layer is used per region and interaction between regions is encoded into the energy function. Besides the usual data term and smoothing term in (2), there is an inter-region interaction term in the energy function in (3).

$$\begin{aligned}
 \text{Energy}(f) = & \\
 & \text{Energy}_{\text{smooth}}(f) [\sum_{p,q \in N} V_{p,q}(f_p, f_q)] + \\
 & \text{Energy}_{\text{data}}(f) [\sum_{p \in F} D_p(f_p)] + \sum_{\substack{m, n \in L \\ m \neq n}} W^{mn}(x^m, x^n)
 \end{aligned} \tag{3}$$

$W^{mn}(x^m, x^n) = \sum_{p,q \in N^{mn}} W_{pq}^{mn}(x_p^m, x_q^n)$ is a geometric term for making interaction between two regions m and n (two superscripts in W), and $x_p^k = 1$ means pixel p is included in region k , $x_p^k = 0$ means pixel p belongs to the background. There are three types of geometric constraints that are applied to the constructed graph: containment, exclusion, and attraction. Containment constraint requires that region B has to be included in region A. Exclusion constraint requires that A and B cannot overlap at any pixel. Attraction requires that region A should not be grown too far away from region B. Figure 3 shows the constraint energy terms between two regions.

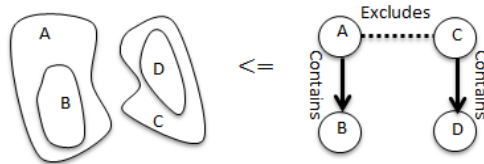


Fig. 3. An example of two mutually exclusive regions [6]

This multi-region model has been successfully applied to medical image segmentation. All graph-cut based segmentation algorithms suffer from some difficulties in the optimization procedure. These difficulties in some cases stem from the maximum flow/minimum cut algorithms which act as the main optimization techniques behind the graph-cut algorithms.

2.3 Maximum-Flow/Minimum-Cut Algorithms

Maximum flow algorithms offer foundations to many global optimization methods. There are many methods for implementing the maximum flow algorithms, such as Goldberg-Tarjan’s [5] “push-relabel” methods and algorithms based on Ford-Fulkerson [7]. In maximum flow algorithms, given a weighted graph where the weight of an edge represents the capacity of the edge, we want to determine the maximum amount of flow that we can send from one vertex S (source) to another vertex T (sink) in the graph, according to the illustration in Figure 1. Locality and parallelization are two important factors in flow propagation in max/min algorithms. Locality refers to fitting the massive and huge graph vertices into computer memory, and parallelization refers to carrying out the steps in maximum flow algorithm asynchronously [5][8]. Absence of each of them will result in an impractical maximum flow algorithm. In addition, the running time of any maximum flow is an

important issue that has to be considered. On one hand, finding the shortest path for sending flow is the main goal. On the other hand, looking for the shortest path is expensive and time consuming.

3 Unified Graphical Models and Image Segmentation

In graph-cut approaches prior knowledge about image data are usually not considered. In order to overcome this weakness, Zhang et al. [1][9] proposed a combinatorial probabilistic framework of a directed (e.g. Bayesian Network) and undirected (e.g. Conditional Random Field (CRF) or Markov Random Model (MRF)) graphical models. It is inspired by human's abilities to segment an image with additional constraints such as contour smoothness, connectivity, object shape, etc. In contrast to graph-cut models which utilize an energy function, probabilistic graphical models perform probabilistic inference for segmentation.

Probabilistic graphical models provide an easy visualization of the structure of a probabilistic model[1][9]. In a probabilistic Graphical Model, each node represents a random variable or group of random variables and the links show probabilistic relationships between these variables. In this regard, the whole graph expresses the way in which the joint distribution over all of the random variables can be decomposed into a product of factors.

Many methods have been proposed for image segmentation based on MRF, CRF, Bayesian Network (BN) and their combination, such as the MRF hybrid model of Hinton et al. [10], the BN and MRF model of Murino et al. [11] and Liu et al. [12]. In the recent method by Zhang and Ji [1], the causal and non-causal relationships among all the image entities are considered. According to Figure 4, it tries to facilitate region-based (upper cloud) and edge-based (lower) segmentation in an image by unifying CRF and BN. This approach consists of three parts, CRF, BN and a combination of these two models. The CRF model (undirected) is constructed based on a superpixel model. Superpixels are specified by an over-segmentation preprocessing task on the image that extracts some homogeneous parts (regions). Each part is a single region node in CRF.

The model is a multi-layer model which supposed the CRF as a labeling system that assigns a label based on local feature vector extracted from image (x) for a superpixel (Y) and BN as an inference system of the type of the edge nodes given various measurements and constraints. CRF part plays the region-based image segmentation role and BN part tackles edge-base segmentation. Because of the inability of CRF in representing causal model, Bayesian Network is deployed to model the causal relationship. BN part provides a representation of the relationships between superpixels (Y), the edge segments (e) and vertices (v). The model has been represented for 4 superpixels in Figure 4. Vertices (v) are a place where three or more edges intersect and M denotes measurements of features of edges (e) and vertices (v).

In the next section, we perform some comparative study to illustrate the performance of the two approaches.

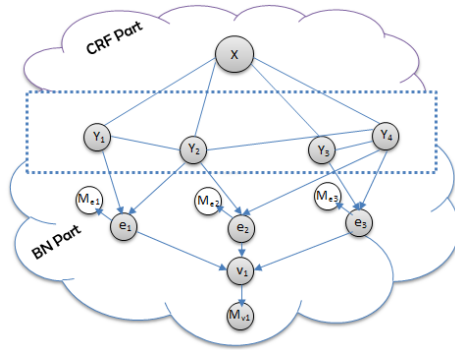


Fig. 4. A structure for unified graphical model combining CRF model and BN model by region nodes [1]

4 Experiments

We perform some segmentation experiments on the Weizmann (horse) dataset [13][14]. Weizmann horse dataset includes color images of 328 different side-views of horses in addition to their ground truth masks. Because of the variety of images with different appearances and side views, this is a challenging dataset for segmentation. We compare the unified graphical model work and multi-label graph-cut [13] on a group of five, 320*213 pixels images of this dataset. Figure 5 shows examples of horse images in color and their binary segmentation results from unified graphical model [1] and graph-cut model [13]. Our results show that for object extraction or foreground/background segmentation, unified graphical model has better performance.



Fig. 5. Comparison between unified graphical model and graph-cut model for the horse dataset-The top row shows original images in color, the middle row is the results of unified graphical model [1]with a 96.6 percent accuracy and the bottom row is the results of graph-cut model[13] with47.8 percent accuracy.

The segmentation accuracy is computed using Dice index as given in (4). In case of the foreground/background we have $n=1$.

$$Dice(R, S) = \frac{1}{n} \sum_{i=1}^n \frac{2|A \cap S_i|}{|A_i| + |S_i|} \quad (4)$$

Here, R denotes the ground truth, S denotes the segmented image and n is the number of segmented areas. The results of graph-cut model show that this model does not have a good performance on images with one object and complicated background. In particular, when there is common color between foreground and background or in the case of the same intensity the model will not show good performance. Unified graphical model shows problems when the background is the same in appearance with the foreground.

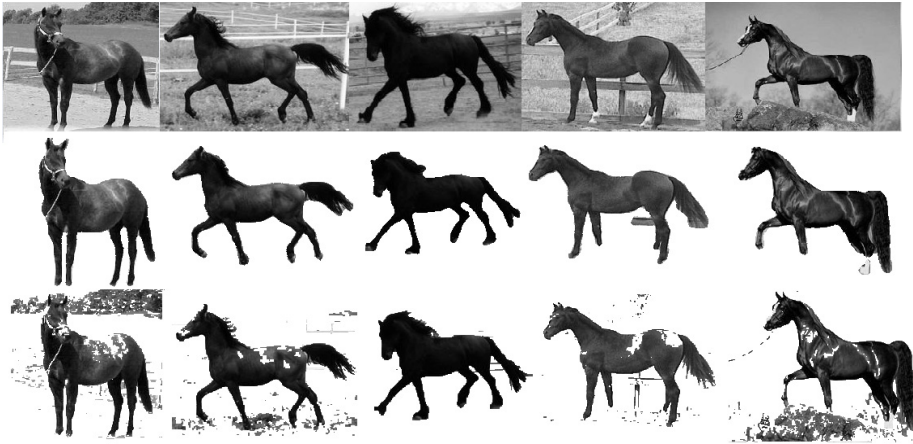


Fig. 6. Comparison between unified graphical model and graph-cut model for the gray-scale horse dataset - The top row shows original images in gray-scale, the middle row is the results of unified graphical model [1] with a 98 percent accuracy and the bottom row is the results of graph-cut model [13] with 91 percent accuracy

We also performed another experimental test on a group of 5 gray-scale images of Weizmann (horse) dataset to show the robustness of these two methods for precise segmentation in absence of color. Figure 6 shows examples of horse images in gray-scale and their binary segmentation results from unified graphical model [1] and graph-cut model [13]. In this case also unified graphical model has better performance than graph-cut model. Nevertheless as it has been shown in Figure 5 and Figure 6 there is a huge difference between graph-cut accuracy results for color images and graph-cut accuracy results for gray-scale ones. In Figure 6 gray-scale dataset images especially the results of graph-cut approach still suffer from intensity problem. Because the main basis of graph-cut approach is based on intensity differences, the performance for images with high difference between foreground and background is much better than the others.

In terms of speed, the speed time is $O(mn^2 |C|)$ [7] for graph-cut, where m and n are the number of edges and nodes, respectively, and $|C|$ is the cost of

minimum cut for graph-cut model. In unified graphical model the time complexity is dependent on the constructed graph. The graph usually consists of 700 to 1500 nodes [1]. For images of the same size, segmentation time complexity for graph-cut model is usually less than that for the unified graphical model.

5 Conclusions

In this paper, we described two graph-based approaches to image segmentation, namely, graph-cut approach and probabilistic graphical modeling approach and compared their performance on several images coming from the commonly used dataset. Our results show that for images with complicated background, the unified graphical model which is able to take into account multiple prior constraints have superior performance.

References

1. Zhang, L., Ji, Q.: Image segmentation with a unified graphical model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 1406–1425 (2010)
2. Boykov, Y., Jolly, M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: *Proceedings of Eighth IEEE International Conference on Computer Vision ICCV 2001*, pp. 105–112 (2001)
3. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision* **70**, 109–131 (2006)
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**, 1222–1239 (2001)
5. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum-flow problem. *Journal of the ACM (JACM)* **35**, 921–940 (1988)
6. Delong, A., Boykov, Y.: Globally optimal segmentation of multi-region objects. In: *IEEE 12th International Conference on Computer Vision*, pp. 285–292 (2009)
7. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1124–1137 (2004)
8. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for ND grids. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pp. 1–8 (2008)
9. Zhang, L., Zeng, Z., Ji, Q.: Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. *IEEE Transactions on Image Processing* **20**, 2401–2413 (2011)
10. Hinton, G.E., Osindero, S., Bao, K.: Learning causally linked markov random fields. In: *AI & Statistics* (2005)
11. Regazzoni, C., Murino, V., Vernazza, G.: Distributed propagation of a-priori constraints in a Bayesian network of Markov random fields. *IEE Proceedings I (Communications, Speech and Vision)* **140**, 46–55 (1993)
12. Liu, F., Xu, D., Yuan, C., Kerwin, W.: Image segmentation based on Bayesian network-Markov random field model and its application to in vivo plaque composition. In: *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 141–144 (2006)
13. <http://www.wisdom.weizmann.ac.il/~bagon>
14. Borenstein, E., Ullman, S.: Learning to Segment. In: *Proc. European Conf. Computer Vision*, pp. 1–8 (2004)

Tolerance Rough Fuzzy Approximation Operators and Their Properties

Yao Zhang¹, Junhai Zhai^{1(✉)}, and Sufang Zhang²

¹ College of Mathematics and Computer Science, Hebei University,
Baoding 071002, China

mczjh@126.com

² Teaching and Research of Section of Mathematics,
Hebei Information Engineering School, Baoding 071000, China

Abstract. In the framework of classification, the rough fuzzy set (RFS) deal with the fuzzy decision tables with discrete conditional attributes and fuzzy decision attribute. However, in many applications, the conditional attributes are often real-valued. In order to deal with this problem, this paper extends the RFS model to tolerance RFS, The definitions of the tolerance rough fuzzy set approximation operators are given, and their properties are investigated.

Keywords: Rough set · Fuzzy set · Rough fuzzy set · Tolerance relation · Approximation operator

1 Introduction

Fuzzy set is usually employed to characterize the uncertainty of cognition[1], while rough set is widely used to describe the uncertainty of knowledge[2]. The typical application of fuzzy set is the fuzzy control[3], while the representative application of rough set is feature selection[4]. In this paper, we discuss the problem of extension of rough set in the framework of classification, i.e. the target concept is the decision class. In this scenario, the combination of fuzzy set and rough set addresses the following three types of situations:

(1) The conditional attributes and decision attribute are all fuzzy.

In this situation, the corresponding rough set model is called fuzzy rough set (FRS). The fuzzy decision tables dealt with by FRS are called fuzzy I-decision tables in this paper. The table 1 is a small fuzzy I-decision table with 6 samples. Where A_1 and A_2 are two fuzzy conditional attributes. A_1 has two fuzzy language terms A_{11} and A_{12} , A_2 has three fuzzy language terms A_{21} , A_{22} and A_{23} . C is a fuzzy decision attribute, which has two fuzzy language terms C_1 and C_2 . The values in the table 1 are fuzzy membership degree of samples belonging to a fuzzy set.

There are 4 FRS models reported in the literature. Based on possibility and necessity theory, in 1990, Dubois and Prade proposed a FRS model[5], which is

Table 1. Fuzzy I-decision table

Samples	A_1		A_2			C	
	A_{11}	A_{12}	A_{21}	A_{22}	A_{23}	C_{11}	C_{22}
x_1	0.9	0.1	0.7	0.2	0.1	0.3	0.7
x_2	0.6	0.4	0.6	0.2	0.2	0.6	0.4
x_3	0.7	0.3	0.0	0.7	0.3	0.4	0.6
x_4	0.3	0.7	0.2	0.7	0.1	0.9	0.1
x_5	0.7	0.3	0.0	0.1	0.9	1.0	0.0
x_6	0.1	0.9	0.0	0.7	0.3	0.5	0.5

the most popular one. Based on the fuzzy inclusion degree, Kuncheva proposed the second FRS model in 1992, and applies this model to feature selection[6]. Also in 1992, based on lattice, Nanda proposed the third FRS model[7]. Based on α -level sets, Yao proposed the fourth FRS model[8]. A comprehensive survey of FRS can be found in [9].

(2) The decision attribute is crisp, the conditional attributes are fuzzy. By now, there is no model in the literature to characterize this problem, we will investigate this model in the future work, we named the corresponding fuzzy decision tables as fuzzy II-decision tables. The table 2 is a small fuzzy II-decision table with 6 samples.

(3) The decision attribute is fuzzy, the conditional attributes are discrete. The corresponding model is called rough fuzzy set (RFS)[5]. In this paper, we extend the RFS model to tolerance RFS (TRFS) by introducing the tolerance rough fuzzy approximation operators. In addition, we also study the properties of the tolerance rough fuzzy approximation operators. The fuzzy decision tables dealt with by TRFS are called fuzzy III-decision tables. The table 3 is a small fuzzy III-decision table with 6 samples.

This paper is organized as follows. Section 2 provides preliminaries. The definition of the tolerance rough fuzzy approximation operators, their properties and the proof are presented in Section 3. Section 4 concluded this paper.

Table 2. Fuzzy II-decision table

Samples	A_1		A_2			C
	A_{11}	A_{12}	A_{21}	A_{22}	A_{23}	
x_1	0.9	0.1	0.7	0.2	0.1	yes
x_2	0.6	0.4	0.6	0.2	0.2	no
x_3	0.7	0.3	0.0	0.7	0.3	yes
x_4	0.3	0.7	0.2	0.7	0.1	no
x_5	0.7	0.3	0.0	0.1	0.9	no
x_6	0.1	0.9	0.0	0.7	0.3	yes

Table 3. Fuzzy III-decision table

Samples	A_1	A_2	C	
			C_1	C_2
x_1	1.5	2.3	0.3	0.7
x_2	3.6	2.7	0.6	0.4
x_3	4.3	5.0	0.4	0.6
x_4	7.3	5.4	0.9	0.1
x_5	3.7	9.1	1.0	0.0
x_6	1.0	2.5	0.5	0.5

2 Preliminaries

In this section, we briefly review the basic concepts, including rough set, rough fuzzy set, and tolerance rough set.

2.1 Rough Set

Rough set (RS) uses a pair of operators to approximate the target concepts. Let $DT = (U, A \cup C)$ be a decision table, where $U = \{x_1, x_2, \dots, x_n\}$, which is a set of n objects, U is usually called a universe. $A = \{a_1, a_2, \dots, a_d\}$ is a set of d attributes used for describing the characteristics of objects. C is class label variable, whose values is in set $d = \{1, 2, \dots, k\}$. In other words, the objects in U are categorized into k classes: U_1, U_2, \dots, U_k . Let $x \in U$ and R is an equivalence relation induced by a subset of A , the equivalence class containing x is given by:

$$[x]_R = \{y | xRy\} \quad (1)$$

For arbitrary target concept, i.e. a decision class $U_i (1 \leq i \leq k)$, the R-lower approximation operator \underline{R} of U_i is defined as follows,

$$\underline{R}(U_i) = \{[x]_R | [x]_R \subseteq U_i\} \quad (2)$$

The R-upper approximation operator \overline{R} of U_i is defined as follows,

$$\overline{R}(U_i) = \{[x]_R | [x]_R \cap U_i \neq \emptyset\} \quad (3)$$

The two-tuple $(\underline{R}(U_i), \overline{R}(U_i))$ is called a rough set.

The universe U can be divided into three disjoint regions by R : positive region $POS(U_i)$, negative region $NEG(U_i)$ and boundary region $BND(U_i)$, where

$$POS(U_i) = \underline{R}(U_i) \quad (4)$$

$$NEG(U_i) = U - POS(U_i) \quad (5)$$

$$BND(U_i) = \overline{R}(U_i) - \underline{R}(U_i) \quad (6)$$

2.2 Rough Fuzzy Set

Rough fuzzy set (RFS) is an extended from rough set by replacing the crisp target concept, i.e. the crisp decision class U_i with a fuzzy target concept, i.e. the fuzzy decision class. For the sake of simplicity, we also use U_i , \underline{R} and \overline{R} to describe the fuzzy decision class, the R-lower approximation operator and the upper approximation operator, respectively. We have the following definition[5].

$$\underline{R}(U_i) = \mu_{\underline{R}(U_i)}([x]_R) = \inf\{\mu_{U_i}(y) | y \in [x]_R\} \tag{7}$$

and

$$\overline{R}(U_i) = \mu_{\overline{R}(U_i)}([x]_R) = \sup\{\mu_{U_i}(y) | y \in [x]_R\} \tag{8}$$

According to the fuzzy extension principle, (7) and (8) can be equivalently written as follows.

$$\underline{R}(U_i) = \mu_{\underline{R}(U_i)}(x) = \inf\{\max(\mu_{U_i}(y), 1 - \mu_R(x, y)) | y \in U\} \tag{9}$$

and

$$\overline{R}(U_i) = \mu_{\overline{R}(U_i)}(x) = \sup\{\min(\mu_{U_i}(y), \mu_R(x, y)) | y \in U\} \tag{10}$$

The two-tuple $(\underline{R}(U_i), \overline{R}(U_i))$ is called a rough fuzzy set.

2.3 Tolerance Rough Set

Tolerance rough set (TRS)[10] is another extension of rough set. TRS extends rough set by replacing a equivalence relation with a similarity relation. The target concept is same as in rough set, which is also a crisp decision class.

Given a decision table $DT = (U, A \cup C)$, R is a similarity relation defined on U , if and only if R satisfies the following conditions:

- (1) Reflexivity, i.e. for each $x \in U$, xRx ;
- (2) Symmetry, i.e. for each $x, y \in U$, xRy , and yRx .

We can define many similarity relations on U , such as[11, 12]:

$$R_a(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{max} - a_{min}|} \tag{11}$$

$$R_a(x, y) = \exp\left(-\frac{(a(x) - a(y))^2}{2\sigma_a^2}\right) \tag{12}$$

$$R_a(x, y) = \max\left(\min\left(\frac{a(y) - (a(x) - \sigma_a)}{a(x) - (a(x) - \sigma_a)}, \frac{(a(x) + \sigma_a) - a(y)}{(a(x) + \sigma_a) - a(x)}, 0\right)\right) \tag{13}$$

where $a \in A$, $x \in U$, and a_{max} and a_{min} denote the maximum and minimum values of a respectively. The σ_a is variance of attribute a . For $\forall R \subseteq A$, we can define the similarity relations induced by subset R as follows.

$$R_\tau(x, y) = \frac{\sum_{a \in R} R_a(x, y)}{|R|} \geq \tau \tag{14}$$

or

$$R_\tau(x, y) = \prod_{a \in R} R_a(x, y) \geq \tau \tag{15}$$

where τ is a similarity threshold.

For each $x \in U$, the τ tolerance class generated by a given similarity relation R is defined as:

$$[X]_{R_\tau} = \{y | (y \in U) \wedge (xR_\tau y)\} \tag{16}$$

The tolerance lower approximation and upper approximation operators are defined as

$$\underline{R}_\tau(U_i) = \{x | (x \in U) \wedge ([X]_{R_\tau} \subseteq U_i)\} \tag{17}$$

and

$$\overline{R}_\tau(U_i) = \{x | (x \in U) \wedge ([X]_{R_\tau} \cap U_i \neq \phi)\} \tag{18}$$

The two-tuple $(\underline{R}_\tau(U_i), \overline{R}_\tau(U_i))$ is called a tolerance rough set.

3 TRFS Approximation Operators and Their Properties

In this section, we present the introduced tolerance rough fuzzy approximation operators and their properties.

3.1 Tolerance Rough Fuzzy Approximation Operators

This paper extends the equivalence relation in rough fuzzy set model to tolerance relation, and the definitions of the tolerance rough fuzzy set approximation operators are given in this section.

Given a fuzzy III-decision table $DT = (U, A \cup C)$, R is a similarity relation defined on U , τ is a similarity threshold, U_i is the i th decision class (i.e. a target concept). The tolerance rough fuzzy lower approximation and tolerance rough fuzzy upper approximation operators are defined as

$$\underline{R}_\tau(U_i) = \mu_{\underline{R}_\tau(U_i)}([x]_{R_\tau}) = inf\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \tag{19}$$

and

$$\overline{R}_\tau(U_i) = \mu_{\overline{R}_\tau(U_i)}([x]_{R_\tau}) = sup\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \tag{20}$$

Similarly to (9) and (10), we have the following equivalent definitions:

$$\underline{R}_\tau(U_i) = \mu_{\underline{R}_\tau(U_i)}(x) = inf\{max(\mu_{U_i}(y), 1 - \mu_{R_\tau}(x, y)) | y \in U\} \tag{21}$$

and

$$\overline{R}_\tau(U_i) = \mu_{\overline{R}_\tau(U_i)}(x) = sup\{min(\mu_{U_i}(y), \mu_{R_\tau}(x, y)) | y \in U\} \tag{22}$$

The two-tuple $(\underline{R}_\tau(U_i), \overline{R}_\tau(U_i))$ is called a tolerance rough fuzzy set.

3.2 The Properties of TRFS Approximation Operators

Given a fuzzy III-decision table $DT = (U, A \cup C)$, R is a similarity relation defined on U , τ is a similarity threshold, U_i and U_j are the i th and j th decision class respectively (i.e. a target concept). The proposed TRFS approximation operators satisfy the following properties:

- (1) $\underline{R}_\tau(U_i) \subseteq U_i \subseteq \overline{R}_\tau(U_i)$
- (2) $\underline{R}_\tau(\phi) = \overline{R}_\tau(\phi) = \phi, \underline{R}_\tau(U) = \overline{R}_\tau(U) = U$
- (3) $\overline{R}_\tau(U_i \cup U_j) = \overline{R}_\tau(U_i) \cup \overline{R}_\tau(U_j)$
- (4) $\underline{R}_\tau(U_i \cap U_j) = \underline{R}_\tau(U_i) \cap \underline{R}_\tau(U_j)$
- (5) $U_i \subseteq U_j \Rightarrow \underline{R}_\tau(U_i) \subseteq \underline{R}_\tau(U_j)$
- (6) $U_i \subseteq U_j \Rightarrow \overline{R}_\tau(U_i) \subseteq \overline{R}_\tau(U_j)$
- (7) $\overline{R}_\tau(U_i \cap U_j) \subseteq \overline{R}_\tau(U_i) \cap \overline{R}_\tau(U_j)$
- (8) $\underline{R}_\tau(U_i \cup U_j) \supseteq \underline{R}_\tau(U_i) \cup \underline{R}_\tau(U_j)$

Proof

(1) According to the definition (19) and (20), we have

$$\underline{R}_\tau(U_i) = \mu_{\underline{R}_\tau(U_i)}([x]_{R_\tau}) = \inf\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\}$$

And

$$\overline{R}_\tau(U_i) = \mu_{\overline{R}_\tau(U_i)}([x]_{R_\tau}) = \sup\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\}$$

Because

$$\inf\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \leq \{\mu_{U_i}(y) | \forall y \in [x]_{R_\tau}\} \leq \sup\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\}$$

Hence

$$\underline{R}_\tau(U_i) \subseteq U_i \subseteq \overline{R}_\tau(U_i)$$

(2) Because

$$\mu_{R_\tau(\phi)}([x]_{R_\tau}) = 0$$

Therefore

$$\begin{aligned} \mu_{\underline{R}_\tau(\phi)}([x]_{R_\tau}) &= \inf\{\mu_\phi(y) | y \in [x]_{R_\tau}\} = \mu_{\overline{R}_\tau(\phi)}([x]_{R_\tau}) \\ &= \sup\{\mu_\phi(y) | y \in [x]_{R_\tau}\} \\ &= 0 \end{aligned}$$

Hence

$$\underline{R}_\tau(\phi) = \overline{R}_\tau(\phi) = \phi$$

Because

$$\mu_{R_\tau(U)}([x]_{R_\tau}) = 1$$

Therefore

$$\begin{aligned} \mu_{\underline{R}_\tau(U)}([x]_{R_\tau}) &= \inf\{\mu_U(y) | y \in [x]_{R_\tau}\} = \mu_{\overline{R}_\tau(U)}([x]_{R_\tau}) \\ &= \sup\{\mu_U(y) | y \in [x]_{R_\tau}\} \\ &= 1 \end{aligned}$$

Hence

$$\underline{R}_\tau(U) = \overline{R}_\tau(U) = U$$

(3) Because

$$\begin{aligned}\overline{R}_\tau(U_i \cup U_j) &= \mu_{\overline{R}_\tau(U_i \cup U_j)}([x]_{R_\tau}) = \sup\{\mu_{U_i \cup U_j}(y) | y \in [x]_{R_\tau}\} \\ &= \max(\mu_{\overline{R}_\tau(U_i)}([x]_{R_\tau}), \mu_{\overline{R}_\tau(U_j)}([x]_{R_\tau})) \\ &= \overline{R}_\tau(U_i) \cup \overline{R}_\tau(U_j)\end{aligned}$$

Hence, the property (3) is hold.

(4) Because

$$\begin{aligned}\underline{R}_\tau(U_i \cap U_j) &= \mu_{\underline{R}_\tau(U_i \cap U_j)}([x]_{R_\tau}) = \inf\{\mu_{U_i \cap U_j}(y) | y \in [x]_{R_\tau}\} \\ &= \min(\mu_{\underline{R}_\tau(U_i)}([x]_{R_\tau}), \mu_{\underline{R}_\tau(U_j)}([x]_{R_\tau})) \\ &= \underline{R}_\tau(U_i) \cap \underline{R}_\tau(U_j)\end{aligned}$$

Hence, the property (4) is hold.

(5)Because

$$\begin{aligned}U_i \subseteq U_j &\Rightarrow \mu_{R_\tau(U_i)}([x]_{R_\tau}) = \{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \leq \mu_{R_\tau(U_j)}([x]_{R_\tau}) \\ &= \{\mu_{U_j}(y) | y \in [x]_{R_\tau}\} \\ &\Rightarrow \mu_{\underline{R}_\tau(U_i)}([x]_{R_\tau}) = \inf\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \leq \mu_{\underline{R}_\tau(U_j)}([x]_{R_\tau}) \\ &= \inf\{\mu_{U_j}(y) | y \in [x]_{R_\tau}\} \\ &\Rightarrow \underline{R}_\tau(U_i) \subseteq \underline{R}_\tau(U_j)\end{aligned}$$

Hence, the property (5) is hold.

(6)Because

$$\begin{aligned}U_i \subseteq U_j &\Rightarrow \mu_{R_\tau(U_i)}([x]_{R_\tau}) = \{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \leq \mu_{R_\tau(U_j)}([x]_{R_\tau}) \\ &= \{\mu_{U_j}(y) | y \in [x]_{R_\tau}\} \\ &\Rightarrow \mu_{\overline{R}_\tau(U_i)}([x]_{R_\tau}) = \sup\{\mu_{U_i}(y) | y \in [x]_{R_\tau}\} \leq \mu_{\overline{R}_\tau(U_j)}([x]_{R_\tau}) \\ &= \sup\{\mu_{U_j}(y) | y \in [x]_{R_\tau}\} \\ &\Rightarrow \overline{R}_\tau(U_i) \subseteq \overline{R}_\tau(U_j)\end{aligned}$$

Hence, the property (6) is hold.

(7) Because

$$\begin{aligned}\overline{R}_\tau(U_i \cap U_j) &= \mu_{\overline{R}_\tau(U_i \cap U_j)}([x]_{R_\tau}) = \sup\{\mu_{U_i \cap U_j}(y) | y \in [x]_{R_\tau}\} \\ &\leq \min(\mu_{\overline{R}_\tau(U_i)}([x]_{R_\tau}), \mu_{\overline{R}_\tau(U_j)}([x]_{R_\tau})) \\ &= \overline{R}_\tau(U_i) \cap \overline{R}_\tau(U_j)\end{aligned}$$

Hence, we have $\overline{R}_\tau(U_i \cap U_j) \subseteq \overline{R}_\tau(U_i) \cap \overline{R}_\tau(U_j)$.

(8) Because

$$\begin{aligned}\underline{R}_\tau(U_i \cup U_j) &= \mu_{\underline{R}_\tau(U_i \cup U_j)}([x]_{R_\tau}) = \inf\{\mu_{U_i \cup U_j}(y) \mid y \in [x]_{R_\tau}\} \\ &\geq \max(\mu_{\underline{R}_\tau(U_i)}([x]_{R_\tau}), \mu_{\underline{R}_\tau(U_j)}([x]_{R_\tau})) \\ &= \underline{R}_\tau(U_i) \cup \underline{R}_\tau(U_j)\end{aligned}$$

Hence, we have $\underline{R}_\tau(U_i \cup U_j) \supseteq \underline{R}_\tau(U_i) \cup \overline{R}_\tau(U_j)$.

4 Conclusions

This paper combines the tolerance rough set and rough fuzzy set. Two TRFS approximation operators are introduced, their properties are investigated, and the proofs of these properties are given. The proposed TRFS approximation operators extended the range of application of the classical rough approximation operators, which can directly deal with the decision table with real value conditional attributes and fuzzy value decision attribute. In the future works, we will study the approximation operators for decision table with fuzzy value conditional attributes and discrete value decision attribute.

Acknowledgments. This research is supported by the national natural science foundation of China (61170040, 71371063), by the key scientific research foundation of education department of Hebei Province (ZD20131028), by the scientific research foundation of education department of Hebei Province (Z2012101), and by the natural science foundation of Hebei Province (F2013201110, F2013201220).

References

1. Zadeh, L.A.: Fuzzy sets. *Inform. Control* **8**, 338–353 (1965)
2. Pawlak, Z.: Rough sets. *International Journal of Information and Computer Sciences* **11**(5), 341–356 (1982)
3. Zimmermann, H.J.: Fuzzy set theory. *Journal of Advanced Review* **2**, 317–332 (2010)
4. Shen, Q., Jensen, R.: Rough Sets, their Extensions and Applications. *International Journal of Automation and Computing* **4**(1), 1–12 (2007)
5. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems* **17**, 191–208 (1990)
6. Kuncheva, L.I.: Fuzzy rough sets: application to feature selection. *Fuzzy Sets and Systems* **51**(2), 147–153 (1992)
7. Nanda, S.: Fuzzy rough sets. *Fuzzy Sets and Systems* **45**, 157–160 (1992)
8. Yao, Y.Y.: Combination of rough and fuzzy sets based on α -level sets. In: Lin, T.Y., Cercone, N. (eds.) *Rough Sets and Data Mining: Analysis for Imprecise Data*, pp. 301–321. Kluwer Academic Publishers, Boston (1997)

9. Yeung, D.S., Chen, D.G., Tsang, E.C.C., et al.: On the generalization of fuzzy rough sets. *IEEE Transactions on Fuzzy Systems* **13**(3), 343–361 (1992)
10. Skowron, A.: Tolerance approximation spaces. *Fundamenta Informaticae* **27**(2–3), 245–253 (1996)
11. Parthaláin, N.M., Shen, Q.: Exploring the boundary region of tolerance rough sets for feature selection. *Pattern Recognition* **42**, 655–667 (2009)
12. Parthaláin, N.M., Shen, Q., Jensen, R.: A distance measure approach to exploring the rough set boundary region for attribute reduction. *IEEE Transactions on Knowledge and Data Engineering* **22**, 306–317 (2010)

Extreme Learning Machine for Interval-Valued Data

Shixin Zhao^{1,2}(✉) and Xizhao Wang³

¹ School of Management, Hebei University, Baoding, Hebei, China
cssxzhao@163.com

² Department of Mathematics and Physics, Shijiazhuang Tiedao University,
Shijiazhuang, Hebei, China

³ Machine Learning Center, Faculty of Mathematics and Computer Science,
Hebei University, Baoding, Hebei, China

Abstract. Extreme learning machine (ELM) is a fast learning algorithm for single hidden layer feed-forward neural networks, but it only can deal with the data sets with numerical attributes. Interval-valued data is considered as a direct attempt to extend precise real-valued data to imprecise scenarios. To deal with imprecise data, this paper proposes three extreme learning machine (ELM) models for interval-valued data. Mid-point and range of the interval are selected as the variables in the first model as in previous works. The second model selects endpoints as variables and produces better performance than model 1. The third model, a constrained ELM for interval-valued data, is built to guarantee the left bound is always smaller than its right bound. Three different standards are used to test the effectiveness of the three models, and experimental results show that the latter two models offer better performances than the former one.

Keywords: Extreme learning machine · Interval-valued · Mid-point of interval · Range of interval · Endpoint of interval

1 Introduction

The capabilities of single-hidden-layer feed-forward neural networks (SLFNs) to approximate complex nonlinear mappings have been widely investigated in various research areas [1–4]. Traditionally, the weights in SLFNs are trained by the gradient-descent based algorithms, one of which is the back-propagation (BP) algorithm. Although Funahashi and Cybenko [5, 6] proved that BP network has the ability to approximate arbitrary continuous functions with arbitrary accuracy, convergence speed of BP algorithm is too slow and easy to fall into the local minimum. Extreme learning machine (ELM) was put forward by Huang in 2006 [7], which is a fast learning algorithm in the framework of SLFNs to overcome the above-mentioned problems such as local minimum and too slow convergence. In ELM, the input weights and hidden layer biases are randomly chosen, while the output weights are analytically determined by generalized Moore-Penrose

inverse operation of the hidden-layer output matrix. Without iterations, ELM can obtain a good performance with very high learning speed in many applications. More and more researchers are attracted by ELMs due to their simplicity and effectiveness. A lot of research achievements have been reported [8–15].

In recent years, the consideration of different sources imprecision in generating and modeling experimental data: uncertainty in the quantification of the data, subjective measurements, perceptions, to name but a few, has greatly promoted and developed the techniques of mining and learning from data. Interval-valued data are considered as a direct attempt to extend real-valued data to more flexible scenarios. For instance, when the characteristic has a great uncertainty in the quantification of its values, it may be reasonable to formalize these values as intervals instead of real numbers [16]. Furthermore, there are essentially interval-valued attributes, as fluctuation of a certain characteristic during a period of time, ranges of a certain variable, and so on [17,18]. Motivated by dealing with uncertainty of data in ELM learning and then improving its learning performance, we propose to extend and adapt the basic ELM to an interval-valued model.

This paper proposed three different interval-valued ELM models. Considering that midpoint and range of the interval are generally set as variables in previous works [19–21], our first model is developed based on midpoint and range of the interval. Unfortunately for this model the experiment result is not satisfactory. In view of importance of the interval endpoint to express the interval information, the second model is formulated based on the endpoint of the interval, which exhibits a relatively better performance than model 1. The second model may cause some abnormal phenomenon that the real output intervals' lower bounds are larger than their upper ones. In order to overcome this defect the third model was established with adding a constraint to guarantee the normal order relation of intervals endpoints, which leads to a satisfactory performance.

This paper is organized as follows. Section 2 gives a simply retrospect of ELM. Three different ELM models for interval-valued data are rigorously proposed in section 3. Performance evaluation is presented in Section 4. Conclusions are given in Section 5.

2 A Brief Review on ELMs

In order to understand the following content better, let's firstly review ELM. ELM is a fast learning algorithm for single hidden layer feedforward neural networks (SLFNs), mathematical model of standard SLFNs with \tilde{N} hidden nodes and activation function $g(x)$ is :

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_j \cdot x_j + b_j) = t_j, \quad j = 1, 2, \dots, N \quad (1)$$

where (x_j, t_j) are N arbitrary distinct samples in database, $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})^T \in R^n$ is input vector of the j -th sample, $t_j = (t_{j1}, t_{j2}, \dots, t_{jm})^T \in R^m$

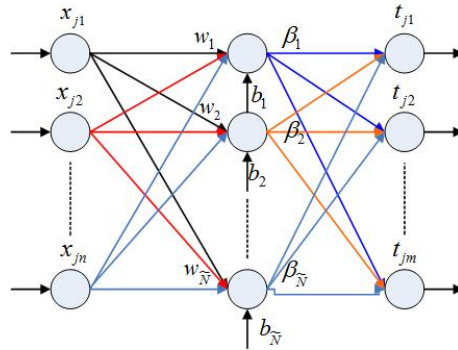


Fig. 1. Structure of SLFNs

is output vector of the j -th sample. $\omega_i = (\omega_{i1}, \omega_{i2}, \dots, \omega_{im})$ is the weight vector connecting the i -th hidden node and the input nodes, $\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})$ is the weight vector connecting the i -th hidden node and the output nodes, and b_i is the threshold of the i -th hidden node. $\omega_i \cdot x_j$ denotes the inner product of ω_i and x_j .

The above N equations can be compactly written as:

$$H\beta = T, \tag{2}$$

where

$$H(\omega_1, \dots, \omega_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) = \begin{pmatrix} g(\omega_1 \cdot x_1 + b_1) & \dots & g(\omega_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(\omega_1 \cdot x_N + b_1) & \dots & g(\omega_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}}, \tag{3}$$

$$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{pmatrix}_{\tilde{N} \times m}, \quad T = \begin{pmatrix} t_1^T \\ \vdots \\ t_N^T \end{pmatrix}_{N \times m}, \tag{4}$$

H is called the hidden layer output matrix of the neural networks. The i -th column of H is output of the i -th hidden node with respect to inputs (x_1, \dots, x_N) .

Dr. Huang proved in [7] that in such a network it is not necessary to adjust all weights. In fact the input weights and bias in hidden layer need not to be iteratively adjusted and it is enough to randomly choose them once at the beginning of training. After this, the hidden layer output matrix H will be fixed and will have no change during the training. Since H is an overdetermined matrix, what we need to do is to look for a least-square solution of equation (2), that is to find the parameter vector $\hat{\beta}$ satisfying the following equality:

$$\begin{aligned} & \left\| H(\omega_1, \dots, \omega_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}) \hat{\beta} - T \right\| \\ & = \min_{\beta} \left\| H(\omega_1, \dots, \omega_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}) \beta - T \right\|. \end{aligned} \tag{5}$$

The smallest norm least-square solution of this above linear system is [22]:

$$\hat{\beta} = H^\dagger T, \tag{6}$$

More references experimentally demonstrate that ELMs can gain an extremely fast learning-speed than previous algorithms and simultaneously have a good generalization performance for both classification and regression problems [7].

3 ELM for Interval-Valued Data

How to deal with the imprecision in the process of generating and modeling experimental or real-world data has attracted the attention of more and more machine learning scholars. As a type of uncertainty and imprecision, the interval-valued data is a direct attempt to extend real-valued data into more flexible scenarios with representation of imprecision. Interval-valued data can express not only the uncertainty mainly caused by data perturbation but also the magnitudes of data fluctuation. Since a basic ELM can deal only with real-valued data and application problems require users to apply ELMs to handle imprecise data, three improved ELM models for interval-valued SLFNs will be set up in this section. The models consider that the both input and the output are interval-values. It is obvious that the three models are extensions of the original ELM in the aspects of input and output data types and corresponding training algorithms since an interval can be regarded as an extension of a real value. It is worth noting that the improved models will be simplified into the original ELM if their real-valued inputs and outputs are provided. The structure of our models is depicted as follows:

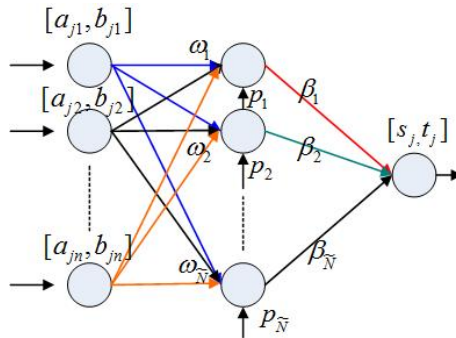


Fig. 2. Structure of an interval-valued SLFN

Three different interval-valued ELM models will be proposed in this section. The first model sets midpoint and range of the interval as variables considering of models in previous works [19–21]. It is a straightforward extension of the existing work and the extension has not an essential difficulty. However we experimentally

find that the performance is not satisfactory. The experimental details can be seen from section 4. In consideration of importance of the interval endpoints, we propose the second model, which is based on the endpoints of the interval. Although the model 2 improves the model 1 to some extent in the aspects of accuracy and robustness, model 2 has shown its own defect. That is the abnormal phenomenon that real output intervals' lower bounds are larger than their upper ones. To overcome this defect, we add a constraint condition in model 2 to guarantee the normal order relation of intervals, and therefore, propose the third model. Similar to the basic ELM, the weights connecting input layer and hidden layer and bias in the hidden layer are set randomly.

Model 1: The real output of the input intervals' midpoint and rang is corresponding equal to target output intervals' midpoint and rang, and Moore-Penrose generalized inverse matrix is used to get the least-square solution of the new model. The mathematical model is:

$$\begin{cases} \sum_{i=1}^{\tilde{N}} \beta_i g_i(\frac{a_j+b_j}{2}) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot (\frac{a_j+b_j}{2}) + p_i) = \frac{s_j+t_j}{2} \\ \sum_{i=1}^{\tilde{N}} \beta_i g_i(\frac{b_j-a_j}{2}) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot (\frac{b_j-a_j}{2}) + p_i) = \frac{t_j-s_j}{2} \end{cases}, \quad j = 1, 2, \dots, N \quad (7)$$

where $([a_j, b_j], [s_j, t_j])$ are N interval-valued samples, $[a_j, b_j]$ is the input interval, $a_j = (a_{j1}, a_{j2}, \dots, a_{jn})^T \in R^n$ is the left point of input interval, $b_j = (b_{j1}, b_{j2}, \dots, b_{jn})^T \in R^n$ is the right point of input interval, so $\frac{a_j+b_j}{2}$ is the midpoint of input interval, $\frac{b_j-a_j}{2}$ is the input interval radius; $[s_j, t_j]$ is the output interval, $s_j = (s_{j1}, s_{j2}, \dots, s_{jm})^T \in R^m$ is the left point of output interval, $t_j = (t_{j1}, t_{j2}, \dots, t_{jm})^T \in R^m$ is the right point of output interval, $\frac{s_j+t_j}{2}$ is the midpoint of output interval, $\frac{t_j-s_j}{2}$ is the output interval radius, $j = 1, 2, \dots, N$; $\omega_i = (\omega_{i1}, \omega_{i2}, \dots, \omega_{in})$ is the weights connecting input layer and hidden layer, $\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})$ is the weights connecting hidden layer and output layer, p_i is the bias in the i -th hidden node, $i = 1, 2, \dots, \tilde{N}$; $\omega_i \cdot (\frac{a_j+b_j}{2})$ is the inner product of ω_i and $\frac{a_j+b_j}{2}$, $\omega_i \cdot (\frac{b_j-a_j}{2})$ is the inner product of ω_i and $\frac{b_j-a_j}{2}$, g is the activation function. Usually the activation function can be taken as sigmoid function. Let

$$\begin{aligned} & \begin{pmatrix} g(\omega_1 \cdot (\frac{a_1+b_1}{2}) + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot (\frac{a_1+b_1}{2}) + p_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \cdot (\frac{a_N+b_N}{2}) + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot (\frac{a_N+b_N}{2}) + p_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}} \\ & \quad \triangleq H_1(\omega_1, \dots, \omega_{\tilde{N}}, p_1, \dots, p_{\tilde{N}}, \frac{a_1+b_1}{2}, \dots, \frac{a_N+b_N}{2}), \\ & \begin{pmatrix} g(\omega_1 \cdot (\frac{b_1-a_1}{2}) + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot (\frac{b_1-a_1}{2}) + p_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \cdot (\frac{b_N-a_N}{2}) + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot (\frac{b_N-a_N}{2}) + p_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}} \\ & \quad \triangleq H_2(\omega_1, \dots, \omega_{\tilde{N}}, p_1, \dots, p_{\tilde{N}}, \frac{b_1-a_1}{2}, \dots, \frac{b_N-a_N}{2}), \end{aligned}$$

$$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{pmatrix}_{\tilde{N} \times m}, T_1 = \begin{pmatrix} \frac{(s_1+t_1)^T}{2} \\ \vdots \\ \frac{(s_N+t_N)^T}{2} \end{pmatrix}_{N \times m}, T_2 = \begin{pmatrix} \frac{(t_1-s_1)^T}{2} \\ \vdots \\ \frac{(t_N-s_N)^T}{2} \end{pmatrix}_{N \times m},$$

Thus equation (7) can be compactly re-written as:

$$\begin{cases} H_1\beta = T_1 \\ H_2\beta = T_2. \end{cases} \tag{8}$$

Let $A = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}$ $B = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}$, then equation (8) can be written as:

$$A\beta = B. \tag{9}$$

Similar as equation (2), the least-square solution of this linear system is:

$$\hat{\beta} = A^\dagger B. \tag{10}$$

Model 2: Let the real outputs of the input interval endpoints equal to the target output interval endpoints. Then, the Moore-Penrose generalized inverse matrix is used to get the least-square solution of this model. The model can be mathematically formulated as:

$$\begin{cases} \sum_{i=1}^{\tilde{N}} \beta_i g_i(a_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot a_j + p_i) = s_j \\ \sum_{i=1}^{\tilde{N}} \beta_i g_i(b_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot b_j + p_i) = t_j \end{cases}, \quad j = 1, 2, \dots, N. \tag{11}$$

where the symbols are defined as same as in model 1. $\omega_i \cdot a_j$ is the inner product of ω_i and a_j , $\omega_i \cdot b_j$ is the inner product of ω_i and b_j . Similar as model 1, let

$$\begin{pmatrix} g(\omega_1 \cdot a_1 + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot a_1 + p_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \cdot a_N + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot a_N + p_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}} \triangleq H_L(\omega_1, \dots, \omega_{\tilde{N}}, p_1, \dots, p_{\tilde{N}}, a_1, \dots, a_N),$$

$$\begin{pmatrix} g(\omega_1 \cdot b_1 + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot b_1 + p_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \cdot b_N + p_1) & \cdots & g(\omega_{\tilde{N}} \cdot b_N + p_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}} \triangleq H_R(\omega_1, \dots, \omega_{\tilde{N}}, p_1, \dots, p_{\tilde{N}}, b_1, \dots, b_N),$$

respectively be hidden layer output matrix of input interval left and right endpoint; $\beta = (\beta_1^T, \beta_2^T, \dots, \beta_{\tilde{N}}^T)^T_{\tilde{N} \times m}$ is the weights matrix connecting hidden layer and output layer, $T_L = (s_1^T, s_2^T, \dots, s_N^T)^T_{N \times m}$ is output matrix of the left endpoint of input interval, $T_R = (t_1^T, t_2^T, \dots, t_N^T)^T_{N \times m}$ is output matrix of the right endpoint of input interval, then equation (10) can be compactly written as:

$$\begin{cases} H_L\beta = T_L \\ H_R\beta = T_R. \end{cases} \tag{12}$$

Let $H = \begin{pmatrix} H_L \\ H_R \end{pmatrix}$ $T = \begin{pmatrix} T_L \\ T_R \end{pmatrix}$ then equation (12) is written as:

$$H\beta = T. \tag{13}$$

The least-square solution of this linear system is:

$$\hat{\beta} = H^\dagger T. \tag{14}$$

We can obtain the weights $\hat{\beta}$ by using equation (14), so as to complete the network training. For any point in the input interval, we can calculate corresponding output using the trained networks. If we want to get the output interval, we can define the corresponding output of input interval's left and right end point as the range of the output interval. At this time, an abnormal phenomenon possibly appears. That is the corresponding output of input interval's left endpoint is larger than the output of the right endpoint. In this case, if we want to obtain the output interval, we must sort the outputs of the input interval's endpoints and set the smaller as left and larger as right.

In order to overcome the defect of model 2, we set up model 3 by adding a constraint in model 2.

Model 3: Let the real outputs of the input interval endpoints equal to the target output interval endpoint with the additional constraint that real output of input interval right endpoint is larger than that of left endpoint. This is quadratic optimization problem with linear constraints. The optimization toolbox LSQlin in Matlab can be used to get the least-square solution of this model which can be mathematically formulated as:

$$\begin{cases} \sum_{i=1}^{\tilde{N}} \beta_i g_i(a_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot a_j + p_i) = s_j \\ \sum_{i=1}^{\tilde{N}} \beta_i g_i(b_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot b_j + p_i) = t_j, \quad j = 1, 2, \dots, N, \\ \sum_{i=1}^{\tilde{N}} \beta_i g_i(a_j) \leq \sum_{i=1}^{\tilde{N}} \beta_i g_i(b_j) \end{cases} \tag{15}$$

Similar to notations in model 2, equation (15) can be compactly written as:

$$\begin{cases} H\beta = T \\ H_L\beta \leq H_R\beta. \end{cases} \tag{16}$$

Calling for the optimization toolbox LSQlin in Matlab, one can get the least-square solution of model 3. And the additional constraint in model 3 guarantees that the corresponding output of input interval's right endpoint is larger than output of the right endpoint, so we can get the output interval by stipulating the corresponding output of input interval's left and right endpoint as the left and right endpoint of the output interval.

4 Performance Evaluation

In this section, the performances of the three proposed models are compared on ten modified real problems in regression areas. The specifications of the original data sets are listed in Table 1. All the simulations are carried out in MATLAB 7.1 environment running in AMD A8-5550M, 2.1GHZ CPU. Ten databases in UCI are modified by giving a random moving around in the vicinity of the original data so as to form interval data. The limit of the random moving can be any value, here we take 0.1. There are 50 hidden nodes assigned for our interval-valued ELM. 50 trials have been conducted for all models and the average results are list in following tables.

Table 1. Real-world regression cases

Data sets	Number of cases	Number of attributes
Airfoil	1503	6
Auto	345	8
Breast	41	10
Computer	209	8
Concrete	1030	9
Fires	517	9
Housing	506	14
Physicochemical	45730	10
Red wine quality	1599	12
Yacht	308	6

The description of three models: midpoint and rang of the interval are set as variables in model 1, and Moore-Penrose generalized inverse matrix is used to get the least-square solution; the left and right endpoint of the interval are set as variables in model 2, Moore-Penrose generalized inverse matrix is also used to get the least-square solution, and we will sort the real output endpoints to get an actual output interval; the left and right endpoint of the interval are also set as variables in model 3, a constraint is added to guarantee to form a interval, and optimization toolbox LSQlin in Matlab is used to get the least-square solution. Detailed descriptions and mathematical models can be found in section 3.

In order to evaluate the three models in multiple perspectives, we set three evaluation standards. The first standard is root-mean-square error (RMSE) of the endpoints of the interval just as in previous works.

The second standard is the proportion of real output of any point in input interval correctly falling into the target output interval (PAPC). For any $p_j \in [a_j, b_j]$, PAPC is calculated as:

$$PAPC = \frac{\sum_{j=1}^N F_j}{N} \tag{17}$$

$$\text{where } F_j = \begin{cases} 1, & \sum_{i=1}^{\tilde{N}} \beta_i g_i(p_j) \in [s_j, t_j], \quad j = 1, 2, \dots, N. \\ 0, & \text{others} \end{cases}$$

The degree of coincidence between real output interval and target output interval is the higher the better, so we set the third standard as the ratio of intersection set and union set of real output interval and target output interval (RIU). Let $[c_j, d_j]$ be the real output interval, then RIU is calculated as:

$$RIU_{\max} = \max_{1 \leq j \leq N} \left(\frac{[c_j, d_j] \cap [s_j, t_j]}{[c_j, d_j] \cup [s_j, t_j]} \right) \tag{18}$$

$$RIU_{\min} = \min_{1 \leq j \leq N} \left(\frac{[c_j, d_j] \cap [s_j, t_j]}{[c_j, d_j] \cup [s_j, t_j]} \right) \tag{19}$$

$$RIU_{\text{avg}} = \frac{\sum_{j=1}^N \frac{[c_j, d_j] \cap [s_j, t_j]}{[c_j, d_j] \cup [s_j, t_j]}}{N} \tag{20}$$

The result of comparison among the three models with respect to the training RMSE is listed in Table 2, from which one can see that the performance of model 1 is acceptable only on a few data sets. It shows a poor performance at most cases. The performances of models 2 and 3 are generally good, and model 2 is slightly better compared with model 3.

Table 2. Comparison among three models on training RMSE of endpoints

Data sets	Model 1	Model 2	Model 3
Airfoil	9.2190 e3±7.2111 e3	0.0367±0.0006	0.0366±0.0006
Auto	1.6123±0.7964	0.0406±0.0006	0.0743±0.1016
Breast	2.3034 e6±3.1334 e6	0.0632±0.0000	0.0632±0.0000
Computer	0.0443±0.0334	0.0105±0.0056	0.0143±0.0250
Concrete	0.1527±0.0355	0.0954±0.0028	0.1700±0.1497
Fires	0.3944±0.2258	0.0608±0.0006	0.0980±0.0858
Housing	0.4472±0.1861	0.1172±0.0017	0.2625±0.2604
Physicochemical	0.6304±0.2889	0.2353±0.0009	0.4732±0.0027
Red wine quality	0.1316±0.0051	0.1267±0.0020	0.2541±0.1967
Yacht	7.2128±3.4708	0.2579±0.0015	0.3453±0.1336

The result of comparison among the three models with respect to the index of training PAPC for any input interval point is shown in Table 3. From table 3 one can find that the performance of model 1 is acceptable only on two data sets (the proportion is acceptable when it is larger than 0.7), model 2 is acceptable on three data sets, and model 3 is acceptable on four data sets. In short, the performance of model 3 is better than the other two.

If we consider RIU is acceptable when it is larger than 0.5, from table 4 one can find that the performance of model 1 is not acceptable on all data sets,

Table 3. Comparison among three models on training PAPC

Data sets	Model 1	Model 2	Model 3
Airfoil	0.5847±0.3748	0.0106±0.0191	0.0092±0.0088
Auto	0.4859±0.1224	0.3385±0.2054	0.4763±0.2476
Breast	0.5956±0.1205	0.0005±0.0034	0.0000±0.0000
Computer	0.7585±0.1332	0.9378±0.1367	0.9181±0.1709
Concrete	0.8462±0.0767	0.7225±0.2446	0.8334±0.1957
Fires	0.5997±0.1278	0.3876±0.2102	0.5264±0.2803
Housing	0.5864±0.1904	0.3427±0.1504	0.5441±0.2979
Physicochemical	0.4506±0.2713	0.1856±0.1879	1.0000±0.0000
Red wine quality	0.5376±0.1116	0.7319±0.2437	0.9272±0.0922
Yacht	0.5299±0.1474	0.2977±0.1311	0.5354±0.3214

Table 4. Comparison among three models on training RIU

Data sets	Model 1			Model 2			Model 3		
	max	min	average	max	min	average	max	min	average
Airfoil	0.0011	0.0000	0.0001	0.9998	0.0000	0.8450	0.9999	0.0000	0.8688
Auto	0.3487	0.0000	0.0565	0.9567	0.0000	0.5289	0.8882	0.0000	0.5701
Breast	0.0009	0.0000	0.0001	1.0000	0.0000	0.8022	1.0000	0.0000	0.8039
Computer	0.8906	0.0158	0.4884	0.9842	0.4707	0.8414	0.9522	0.4392	0.8086
Concrete	0.8486	0.0000	0.2516	0.9710	0.0000	0.3470	0.7811	0.0000	0.3061
Fires	0.7398	0.0000	0.1944	0.9799	0.0000	0.4894	0.8293	0.0000	0.4677
Housing	0.6564	0.0000	0.1372	0.9686	0.0000	0.3679	0.7406	0.0000	0.3052
Physicochemical	0.7416	0.0000	0.0745	0.9633	0.0000	0.0738	0.0000	0.0000	0.0000
Red wine quality	0.9229	0.0000	0.1418	0.9531	0.0000	0.2165	0.6864	0.0000	0.2131
Yacht	0.3653	0.0000	0.0324	0.9035	0.0000	0.1442	0.6614	0.0000	0.1084

model 2 and model 3 are all acceptable on four data sets, and the performance of model 3 is slightly than model 2. In short, the performances of model 2 and model 3 are better than model 1.

From what has been discussed above, the performances of model 2 and model 3 are much better than model 1, and model 3 is much better than model 2 when considering the sorting. So model 3, the constraint interval-valued ELM, is an effective model to deal with interval-valued SLFNs.

5 Conclusions

This paper proposes three ELM models for interval-valued data, which extend the original ELMs from handling real-valued attributes to handling interval-valued attributes. This extension will have a significant impact on the application ability of ELM to deal with uncertain data.

Similar to the traditional works on interval-valued data, model 1 is set up with midpoint and range of the interval. Experimental results show that model 1's performances are not acceptable on almost all data sets under the three standards.

Model 2 is established based on the endpoints of the interval. At most cases model 2 can achieve a comparable better performance than model 1. However there exists an inherent defect in model 2, that is, the value of left endpoint for the outputted interval is possibly bigger than the value of right endpoint, which indicates that the real output does not form an interval.

Model 3 is developed in order to overcome the defect of model 2. A constraint which is formulated by the value of left endpoint being less than or equal to the value of right endpoint for the all outputs is added in model 3. Mathematically the model changes from a non-constraint optimization problem to an optimization problem with a quadratic objective function and linear constraints.

We have experimentally tested the learning and prediction accuracies of the three models on ten modified UCI data sets. Experimental results show that, in comparison with models 1 and 2, model 3 wins at most cases. Model 3 can yet be regarded as a kind effective method to deal with interval-valued data.

Our future work on this topic will include further investigating the generalization ability of interval-valued ELMs and applying these models to learning from big data with the focus on analyzing the uncertainty represented in big data and developing the classification and regression techniques for big data.

Acknowledgments. This work is supported in part by Natural Nature Science Foundation of China (No. 61170040, 71371063, 71201111).

References

1. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford Univ. Press, New York (1995)
2. Hagan, M.T., Demuth, H.B., Beale, M.H.: *Neural Network Design*. PWS Publishing, Boston (1996)
3. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Macmillan, New York (1994)
4. Looney, C.G.: *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*. Oxford Univ. Press, New York (1997)
5. Funahashi, K.I.: On the Approximate Realization of Continuous Mapping by Neural Networks. *Neural Networks* **2**, 183–192 (1989)
6. Cybenko, G.: Approximation by Superpositions of a Sigmoid Function. *Mathematics of Control Signals and Systems* **2**(4), 303–314 (1989)
7. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: Theory and applications. *Neurocomputing*, 489–501 (2006)
8. Liu, Q., He, Q., Shi, Z.-Z.: Extreme support vector machine classifier. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) *PAKDD 2008*. LNCS (LNAI), vol. 5012, pp. 222–233. Springer, Heidelberg (2008)
9. Frnay, B., Verleysen, M.: Using SVMs with randomised feature spaces: An extreme learning approach. In: *Proc. 18th ESANN*, Bruges, Belgium, pp. 315–320 (April 2010)
10. Huang, G.-B., Ding, X.-J., Zhou, H.-M.: Optimization method based extreme learning machine for classification. *Neurocomputing* **74**(1–3), 155–163 (2010)
11. Huang, G.-B., Chen, L.: Convex incremental extreme learning machine. *Neurocomputing* **70**(16–18), 3056–3062 (2007)

12. Huang, G.-B., Chen, L.: Enhanced random search based incremental extreme learning machine. *Neurocomputing* **71**(16–18), 3460–3468 (2008)
13. Huang, G.-B., Chen, L., Siew, C.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **17**(4), 879–892 (2006)
14. Toh, K.A.: Deterministic neural classification. *Neural Computing* **20**(6), 1565–1595 (2008)
15. Huang, G.-B., Zhou, H.-M., Ding, X.-J., Zhang, R.: Extreme learning machine for regression and multi-class classification. *IEEE Transaction on System, Man, and Cybernetics-Part B: Cybernetics* **42**(2), 513–529 (2012)
16. Jahanshahloo, G.R., Hosseinzadeh, L.F., Rostamy, M.M.: A generalized model for data envelopment analysis with interval data. *Applied Mathematical Modelling* **33**, 3237–3244 (2008)
17. Diamond, P.: Least squares fitting of compact set-valued data. *Journal of Mathematical Analysis and Applications* **147**, 531–544 (1990)
18. Gil, M.A., Lubiano, A., Montenegro, M., L’opez-Garc’ia, M.T.: Least squares fitting of an affine function and strength of association for interval-valued data. *Metrika* **56**, 97–111 (2002)
19. Blanco Fernandez, A., Colubi, A., Gonzalez-Rodríguez, G.: Linear Regression Analysis for Interval-Valued Data Based on Set Arithmetic: A Review. *Towards Advanced Data Analysis by Combining Soft Computing and Statistics* 285, 19–31 (2013)
20. Neto, E.A.L., de Carvalho, F.A.T., Bezerra, L.X.T.: Linear regression methods to predict interval-valued data. In: Ninth Brazilian Symposium on Neural Networks, SBRN 2006, pp. 125–130 (2006)
21. Maia, A.L.S., de A.T. de Carvalho, F.: Fitting a least absolute deviation regression model on interval-valued data. In: Zaverucha, G., da Costa, A.L. (eds.) SBIA 2008. LNCS (LNAI), vol. 5249, pp. 207–216. Springer, Heidelberg (2008)
22. Serre, D.: *Matrices. Theory and Applications*. Springer, New York (2002)

Credibility Estimation of Stock Comments Based on Publisher and Information Uncertainty Evaluation

Qiaoyun Qiu, Ruifeng Xu^(✉), Bin Liu, Lin Gui, and Yu Zhou

Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School
Harbin Institute of Technology, Harbin, China
qiuqiaoyun@outlook.com, {xurui.feng.hitsz, guilin.nlp,
zhouyu.nlp}@gmail.com

Abstract. Recently, there are rapidly increasing stock-related comments sharing on Internet. However, the qualities of these comments are quite different. This paper presents an automatic approach to identify high quality stock comments by means of estimating the credibility of the comments from two aspects. Firstly, the credibility of information source is evaluated by estimating the historical credibility and industry-related credibility using a linear regression model. Secondly, the credibility of the comment information is estimated through calculating the uncertainty of comment content using an uncertainty glossary based matching method. The final stock comment credibility is obtained by incorporating the above two credibility measures. The experiments on real stock comment dataset show that the proposed approach identifies high quality stock comments and institutions/individuals effectively.

Keywords: Credibility estimation · Information uncertainty · Information source credibility

1 Introduction

In recent years, with the rapid development of Internet, it is becoming the most important information source. In the financial field, many investment institutions and individuals share their stock market related comments and suggestions on Internet. Such information is expected to be helpful to an investor as the reference of investment. However, the quality and credibility of these abundant comments and suggestions are quite different. How to identify the convincing stock comments on the Internet is still a problem, especially for an amateur stock investor.

The mass information are generated and transmitted every day while essential ones are mixed with many useless and even low-quality ones. It motivates the research on information credibility estimation which aims to identify credible and high quality information. There are some works on information credibility estimation. The main stream works focus on the credibility estimation of information publisher through estimate of their influence. The information shared by high influence publisher is regarded as with high quality.

In our opinion, the information credibility comes from two major factors, namely the credibility/confidence of information publisher and the certainty of the information. Take the credibility estimation of stock predictions, which is an important topic in the finance analysis field, as example. A high quality prediction comment should be issued by a high credibility publisher with less uncertainty. The problem of information credibility estimation then becomes two sub-problems: 1. How to estimate the credibility/confidence of institutions/individuals as publisher; and 2. How to estimate the uncertainty of the comments. To this end, we estimate the credibility of stock comments from two aspects:

(1) Information source (name as publisher) credibility In order to estimate the credibility of a publisher, we verify the stock comments proposed from the publisher by comparing the consistency between their predicated stock price and corresponding real stock prices. Naturally, the publishers with better historical consistency (“History Value”) are more convincing. In addition, considering that different investors always focus on different fields, we think that each publisher has different background on different industries. Therefore, a measurement, called as “Industry Value”, is proposed to measure the credibility of publisher in specific industry. Here, the linear regression model is adopted. The history value and industry value are incorporated to generate the information source credibility.

(2) The uncertainty of information

In some comments, the publishers give their comments with different certainty level. Naturally, the comment with less uncertainty should be more convincing. Thus, we proposed a metric, called “Uncertainty Value”, to measure the credibility of the information content itself. Here, uncertainty glossary based analysis method is applied.

The final credibility of stock comment is obtained by incorporating the estimation of publisher credibility and information uncertainty. Evaluations on a stock comments dataset show that the proposed credibility estimation approach identifies the high quality comments and institutions/individuals effectively.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 presents our credibility estimation approach. Section 4 gives the evaluation and discussion. Finally, Section 5 concludes.

2 Related Works

The study on information credibility began in the field of psychology and social science. DiFonzo [1] analyzed rumors in society from the perspective of social psychology. In recent years, Internet is becoming the most important source of information generation and information sharing. Thus, information credibility research attracts widespread attention in information field, especially in the field of computational linguistics and natural language processing.

2.1 Information Credibility Estimation

Ratkiewicz established a classifier using intrinsic features (such as links and labels) on Twitter to identify politically relevant information on Twitter and estimate their cred-

ibility [2]. Qazvinian [3] split the detection of false information into two sub-tasks, namely false information detection and classification of user attitude. The features in text content (such as 2-gram) and the inherent features from Twitter are incorporated in a Bayesian classifier to identify the false information.

Morris investigated the factors for user credibility estimation through questionnaire analysis [4]. This study shows that credibility rating given by the user and the authenticity of event are not directly related. In addition, the microblog topic plays an important role in credibility rating. Castillo's investigated the information scoring method for news topics [5] through analysis of hot events on Twitter. Experimental results show that the appearance of the hyperlink address is an important feature for false information detection. Meanwhile, the high reliability information is usually posted by users with a large spread of history and more forwarders.

2.2 The Publisher Credibility Calculation

Agichtein studied the user's credibility on Yahoo Answers [6]. By analyzing the interactions between users on Yahoo Answers, all users are mapped into a directed graph. Using the user rating information, the credibility of users in the community is estimated. Weng proposed a Tweet Rank model to measure user credibility on Twitter with the comprehensive consideration of the correlation between topics and social structure of the network among users [7]. Pal et al. classified the users on Twitter into two categories, namely authoritative users and non-authoritative users, by using a probabilistic clustering model with the features of user information and user top-1 follower [8]. E. Bakshy[9] estimated the user credibility by calculating the influence of user's micro-blogs. For micro-blogs contained hyperlinks addresses issued by each user, the transmission of hyperlink address in the network are traced to extract the information flow for estimating the influence of micro-blog. Ultimately, the user credibility is estimated through weighting the influence of the user who posted the micro-blogs.

2.3 Uncertainty Analysis

There existed some researches on uncertainty analysis in scientific texts. Based on the thesis collection on biological science and technology, Medlock and Briscoe developed a semi-supervised learning model with text features, such as Tagging, lemmatization and bigram, for text uncertainty analysis [10]. Based on their work, Szarvas [11] proposed a text uncertainty analysis method using Maximum Entropy classifier with bigram and trigram features.

In CoNLL shared task of 2010, indeterminate sentence recognition was proposed as a separated task. The mission is to find uncertain sentences from Wikipedia articles and biology texts.

3 Our Approach

In our opinion, the confidence of a comment comes from two aspects: the credibility of publisher who gives the comment and the certainty of the comment content. Accordingly, we investigate the credibility estimation methods as follows:

(1) Information source (namely publisher) credibility Firstly, the stock comments from target publishers are collected as well as the relevant historical stock price. Using the historical stock price as the evidence, the prediction accuracy of the target publisher is obtained by using a linear regression model. Based on this, the historical credibility value, called “History Value” for each publisher is obtained.

To solve the problem of imbalanced number of comments by different publisher, Confidence interval is introduced to re-weight the History Value. Generally speaking, the credibility of publisher which posts few comments will be decreased.

In addition, the institution/individual may be skilled in a certain field. Thus, a publisher normally has a relatively rich knowledge on some special industries. We classify the stock comments into different rich industries and estimate the credibility of publishers in each industry, respectively. In this way, the “Industry Value” of publisher is obtained.

(2) The uncertainty of comment information

Comments are written with different levels of confidence by using the certainty or uncertainty expressions.

For example:

E.g. 1 “我听说微软可能会收购诺基亚”

(I heard that Microsoft will probably purchase Nokia)

E.g. 2 “微软宣布将收购诺基亚”

(Microsoft declared that they will purchase Nokia)

Obviously, the two sentences are written in different uncertainty level. Naturally, the comment with less uncertainty should be more convincing. This indicates that the information uncertainty should be considered in comment credibility estimation.

Here, we employ the vector space model to present the information content. An uncertainty glossary based method is employed to generate the measurement, called “Uncertainty Value”.

3.1 System Overview

The overview of this system is illustrated in Figure 1.

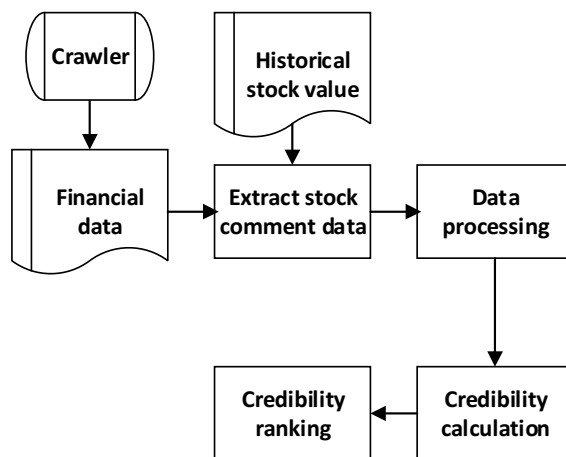


Fig. 1. System overview

Firstly, the stock market comment data is collected from financial websites (such as stock.hexun.com). The target information are then extracted from the crawled un-structured data, such as date, publisher, stock code, and target price etc.

By comparing the actual stock price in history and the prediction posted by publisher, the system estimates the confidence of each publisher in three aspects:

- (1) History Value
- (2) Industry Value
- (3) Uncertain Value

The final credibility is then applied to ranking the credibility of stock comment.

Here, the stock rating mainly includes six levels, define as, “买入(*buy*)”, “增持(*overweight*)”, “持有(*hold*)”, “中性(*neutral*)”, “减持(*underweight*)”, “沽售(*selling*)”.

3.2 Estimation of Information Source Credibility

5,000 stock comments are crawled from financial websites. The corresponding information including date, publisher, stock code, target price, stock ranting, uncertain words and stock prices for the following 7 days, are extracted.

(1)History Value

The ordinary linear regression model is applied to calculate the predicating accuracy of the publishers.

Given the real stock prices of the following seven days corresponding to one stock comment as illustrated below, the real stock line can be obtained.

Table 1. An example of stock price data

Date	12/2	12/3	12/4	12/5	12/6	12/9
Price	3.13	3.14	3.14	3.08	3.13	3.09

Using these data, the linear regression algorithm is employed to grade the stock by converting the gradient of linear regression into a threshold based method and shown as follow:

Table 2. Stock grading strategy

Angle	90~30	30~10	10~1	1~-1	-1~-10	-10~-90
Rating	Buy	Over-weig ht	Hold	Neutral	Un- der-weight	Selling

By comparing the predicated grading by publisher and the actual grading results, the history value of information source credibility is obtained.

(2)Industry Value

The whole stock market is divided into 16 different industries such as mining, manufacturing, financial services, etc. which is suggested Shenzhen Stock Exchange (www.szse.cn). We further estimate the confidence of information sources in separate

industry because the publisher tends to be professional in limited industries. These confidences are obtained by following the above linear regression method. They are regarded as Industry Value.

3.3 Estimation of Information Content Credibility

The information content credibility estimation is mainly based on the calculation of uncertainty of the stock comments. Here, Uncertainty refers to the situation that can't describe accurately of a state or event because of lack of existing knowledge or evidence.

Here, an uncertainty glossary is manually established as shown in Table 3.

Table 3. Uncertainty dictionary

Uncertainty glossary
可能(<i>possible</i>), 大概(<i>probably</i>), 或许(<i>perhaps</i>), 也许(<i>maybe</i>), 听说(<i>heard</i>), 宣称(<i>declare</i>), 肯定(<i>sure</i>), 必然(<i>certain</i>), 必定(<i>certainly</i>), 一定(<i>must be</i>), 定然(<i>definitely</i>), 坚信(<i>firmly believe</i>), 相信(<i>believe</i>), 确信(<i>assurance</i>), 看好(<i>promising</i>), 强烈推荐(<i>highly recommend</i>), 强推(<i>highly recommend</i>), 谨慎推荐(<i>cautious recommendation</i>), 审慎推荐(<i>prudent recommendation</i>), 强买(<i>strong buy</i>), 强烈买入(<i>strong buy</i>), 最看好(<i>the most promising</i>)...

A dictionary matching based method is employed to estimate the matching rating between stock comments and uncertainty glossary.

The following is an example sentence:

3八角项目盈利值得期待，强烈推荐评级

(*Octagonal project profitability worth the wait. Give it the rating of strongly recommended*)

In this example, one word “强烈推荐” is matched. The corresponding pre-defined uncertain value, in this case 0.0, is assigned as the comment content uncertainty value.

3.4 Process of Imbalanced-Comments Publisher

In this study, the history value is determined by the historical prediction consistency of a publisher. The credibility is defined as follows:

$$\text{Credibility value} = \frac{\# \text{the sum of credibility value}}{\# \text{the total number of comments}} \quad (1)$$

The higher value means the higher credibility. However, this value is not always fair. For the two example publishers shown in Table 4, the publisher who provides less comments is assigned a higher credibility value.

Table 4. Credibility of two example publishers

Publisher	Total credibil- ity	Comment's number	Average cred- ibility
P1	43	5	8.6
P2	167.5	25	6.7

To solve this problem, we introduce the concept of confidence interval from statistics to reweight the credibility. Here, we assume the prediction of publisher follows the Gaussian distribution. For each publisher, we utilize the historical predictions to calculate and, fulfils:

$$\frac{x-\mu}{\sigma} \sim N(0,1) \tag{2}$$

For this normal distribution, we observe how to estimate the worst case one publisher achieved. If the worst case of one publisher is still better than others, we have high confidence that this publisher should have a higher rank. Here, we utilize the significant level of 0.05 to estimate the worst case. This means that the prediction of one publisher is higher than this case with the probability 95%.

$$\int_{-\infty}^{u_a} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}} dx = 0.05 \tag{3}$$

Finally, we obtain the credibility of publishers equals to $\bar{x} - \mu_a \frac{\sigma}{\sqrt{n}}$. We only consider the unilateral quantile because we want to know the worst case. If even in the worst case, the publisher is still credible, we believe that the publisher is credible.

4 Experiments and Results

4.1 The Credibility of Stock Comment

We collected 36,588 stock comments and selected 5,829 sets of data (18,424 stock comments) to evaluate our approach. Each group contains stock comment from at least two publishers for the same stock on the same date. We also get the stock data for 7-days following the time it is commented by the two publishers.

Before verification, our system returns a publishers' credibility rating. For each set, we compare the real data with the rating results which is the highest ranked by our system. If the rating fit the real data, we believe that in this set, our rating result is correct. Counting the number of correct results and then calculating the precision as follow.

The evaluation metric is shown as following:

$$P = \frac{\#extracted\ correctly}{\#total\ data} \tag{4}$$

In this experiment, four methods were compared. In the first method, we just consider the History Value. In the second and third methods, we independently added Industry

Value and Uncertain Value, respectively. In the last method, all the three values were applied to evaluate the publisher credibility. The results are showed in Table 5:

Table 5. Performance on publisher credibility estimation

Method	Extracted correctly	total data	precision
HV	3614	5829	62.00%
HV+UV	3788	5829	64.99%
HV+IV	4022	5829	68.99%
HV+UV+IV	4314	5829	74.01%

Here, HV refers to History Value, UV refers to Uncertain Value, and IV refers to Industry Value. The results show that our method is able to return valuable stock comments at precision higher than 74%.

4.2 Uncertainty Word Extraction

In this experiment, we selected 1000 annotated comments to evaluate the uncertainty word extraction. It contains two kinds of annotated results: “uncertainty word” and “none”. “None” means that the data contains no uncertainty words. Our system submits the data which has uncertainty words only.

For the evaluated metric, we utilize the precision, recall and F1 measure, the precision is defined as:

$$P = \frac{\#extracted\ correctly}{\#total\ data} \quad (5)$$

Recall is defined as:

$$R = \frac{\#extracted\ correctly}{\#total\ data} \quad (6)$$

And F1 measure is defined as:

$$F1 = \frac{2P \times R}{P + R} \quad (7)$$

In our experiment, 756 uncertainty cases are identified, in while 687 cases are correct. The performance is shown in Table 6:

Table 6. Performance of uncertainty word extraction

F1	Precision	Recall
78.24%	90.87%	68.7%

Generally speaking, our extraction strategy is a rule based method, such that it achieves a high precision with low recall. Because of the high accuracy dictionary we built, our method could achieve high precision by the rule based method. However, it is very hard to cover all situations among language expression under this strategy. For this reason, the recall of our method is limited.

5 Conclusion

This paper presents an approach to estimate the credibility of a certain stock comment. In our approach, we proposed to estimate the credibility of a comment from two aspects, namely publisher and uncertainty in information content, respectively. The incorporation of these two credibilities becomes the comment credibility. The evaluation on a stock comment dataset shows that the proposed approach extracts the high quality comment from the internet information with a good precision.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 61300112, 61370165), Natural Science Foundation of Guangdong Province (No. S2012040007390, S2013010014475), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, MOE Specialized Research Fund for the Doctoral Program of Higher Education 20122302120070, Open Projects Program of National Laboratory of Pattern Recognition, Shenzhen International Cooperation Research Funding GJHZ20120613110641217, and Shenzhen Foundational Research Funding JCYJ20120613152557576 and JC201005260118A.

References

1. Hovland, C.I., Weiss, W.: The influence of source credibility on communication effectiveness. *Public Opinion Quarterly* **15**(4), 635–650 (1951)
2. Ratkiewicz, J., Conover, M., Meiss, M., et al.: Detecting and Tracking the spread of Astroturf Memes in Microblog streams. arXiv preprint arXiv:1011.3768 (2010)
3. Qazvinian, E., Rosengren, D.R., Mei, Q.: Rumor has it: identifying misinformation in microblogs. In: *Proceedings of EMNLP*, pp. 1589–1599 (2011)
4. Morris, M.R., Counts, S., Roseway, A., et al.: Tweeting is believing?: understanding microblog credibility perceptions. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pp. 441–450 (2012)
5. Castillo, M.M., Poblete, B.: Information credibility on twitter. In: *Proceedings of International Conference on World Wide Web*, pp. 675–684 (2011)
6. Agichtein, E., Castillo, C., Donato, D., et al.: Finding high-quality content in social media. In: *Proceedings of the International Conference on Web Search and Web Data Mining*, pp. 183–194 (2008)
7. Weng, J., Lim, E.P., Jiang, J., et al.: Twitter rank: finding topic-sensitive influential twitterers. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 261–270. ACM (2010)
8. Pal, A.: Counts S.. Identifying topical authorities in microblogs. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining ACM*, pp. 45–54 (2011)
9. Bakshy, E., Hofman, J.M., Mason, W.A., et al.: Everyone’s an influencer: quantifying influence on twitter. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 65–74. ACM (2011)
10. Medlock, Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: *Proceedings of ACL* (2007)
11. SzarvasG.: Hedge classification in biomedical texts with a weakly supervised selection of keywords. In: *Proceedings of ACL* (2008)

A Fast Algorithm to Building a Fuzzy Rough Classifier

Eric C.C. Tsang^{1(✉)} and Suyun Zhao²

¹ The Macau University of Science and Technology, Taipa, Macau, China

² Key Laboratory of Data Engineering and Knowledge Engineering,
Renmin University of China, MOE, Beijing 100872, China
cctsang@must.edu.mo

Abstract. In this paper, by strict mathematic reasoning, we discover the relation between the similarity relation and lower approximation. Based on this relation, we design a fast algorithm to build a rule based fuzzy rough classifier. Finally, the numerical experiments demonstrate the efficiency and the affectivity of the proposed algorithm.

Keywords: Fuzzy rough sets · Rule based classifier · Similarity relation

1 Introduction

Rough set (RS) theory, proposed by Pawlak 910, is a mathematical tool to handle uncertainty of indiscernibility. It is effective in many real applications, such as artificial intelligence, data mining and pattern recognition. However, it is limited by its basic definition ‘equivalence relation’. As a result, many generalizations have been proposed, Fuzzy rough sets 678, Cover rough sets 1112 and so on. These generalizations make rough sets feasible to handle many types of practical problems, such as the problems with real values, the problems with missing values, etc.

The rough classifier has been proposed in recent decades. Roughly speaking, there are two approaches of designing rough/fuzzy rough classifiers. One approach of fuzzy rough classifiers is to induce some if-then rules, whereas the other approach is to compute lower approximation. Rule based fuzzy rough classifier is proposed in [13][15]. These pieces of work proposed a new concept named consistence degree. Based on these concepts, some redundant attribute values can be reduced and then some rules are induced. Furthermore, the authors in [13][15] proposed a rule covering system. Based on this system, a rule based classifier is then designed. Another approach of fuzzy rough classifiers is to compute the lower approximation. Based on this lower approximation, the decision class of the new coming objects is then predicated. The differences between these two types of classifiers are obvious. Firstly the rule based classifier needs training, whereas the lower approximation based classifier needs no training. Secondly the rule based classifier is easy to understand, whereas the lower approximation is not. Thirdly the rule based classifier is complex to execute, whereas the lower approximation based classifier is easy to execute.

In this paper we focus on studying the rule based classifier. We find that it is a little time consuming to build the rule based fuzzy rough classifier. Based on a series of

mathematical analysis we find that the maximum similarity is very important in a rule based classifier. We then proposed a fast algorithm which can build the fuzzy rough classifier quickly.

The remainder of this paper is then organized as follows. Section 2 reviews some preliminary studies on fuzzy rough sets and fuzzy rough classifiers. Section 3 analyzes some notions. In Section 4, a fast algorithm to build fuzzy rough classifiers is designed based on some basic notions. Finally, the numerical experiments in Section 5 demonstrate the effectiveness and efficiency of the proposed algorithm. Section 6 concludes this paper.

2 Preliminaries

In this section, fuzzy rough sets and fuzzy rough classifiers are briefly reviewed. These reviews clearly show that the proposed method in this paper is novel and necessary in real applications.

2.1 Fuzzy Rough Sets

Let $U = \{x_1, x_2, x_3, \dots, x_n\}$, called the Universe, be a nonempty set with a finite number of objects. Each object is described by a set of condition attributes, denoted by $R = \{r_1, r_2, \dots, r_k\}$, and the decision attributes, denoted by D . The triple of $(U, R \cup D)$ is then called a decision system, denoted by DS. With every $P \subseteq R$, we associate a fuzzy binary relation $P(x, y)$ called the fuzzy similarity relation of P , which is a fuzzy binary relation. For simplicity, P is used to represent its similarity relation.

The existing fuzzy rough sets can be summarized as follows:

$$\overline{R}_T A(x) = \sup_{u \in U} T(R(x, u), A(u)) \tag{1}$$

$$\underline{R}_S A(x) = \inf_{u \in U} S(N(R(x, u)), A(u)) \tag{2}$$

$$\overline{R}_\sigma A(x) = \sup_{u \in U} \sigma(N(R(x, u)), A(u)) \tag{3}$$

$$\underline{R}_\vartheta A(x) = \inf_{u \in U} \vartheta(R(x, u), A(u)). \tag{4}$$

To make this paper easy to understand and follow, we take $\underline{R} A(x) = \inf_{u \in U} \min(1 - R(x, u) + A(u), 1)$ and $\overline{R} A(x) = \sup_{u \in U} \max(R(x, u) + A(u) - 1, 0)$ as specific cases of fuzzy rough sets.

2.2 Fuzzy Rough Classifiers

A fuzzy rough classifier is proposed based on the discernibility vector [13]. There are three main highlights.

First, a concept called the consistence degree is proposed, which is the boundary value to keep the information invariant before and after value reduction. The consistence degree of x is defined as $\underline{R}_\beta A(x) = \inf_{D(x,u)=0} \theta(R(x,u), A(u))$.

It is easy to see that this consistence degree in fact is the lower approximation value of x on its decision class. This consistence degree is the minimum distance to the different classes.

Second, the α -discernibility vector is designed to compute the discernibility power of a fuzzy decision table. Each entry of this discernibility vector is designed as $m_j = \{r: T(r(x, x_j), \underline{R}_\beta [x]_D(x)) = 0\}$ for $D(x, x_j) = 0$. For the specific case mentioned in Subsection 2.1, this vector degenerates into $m_j = \{r: \max(r(x, x_j) + \lambda - 1, 0) = 0\}$ for $D(x, x_j) = 0$, where $\lambda = \underline{R}_{\beta_{TL}} A(x)$. Let $d_r(x, x_j) = 1 - r(x, x_j)$, $m_j = \{r: \max(\lambda - d_r(x, x_j), 0) = 0\}$. This discernibility vector shows that if the difference between the distance between x and x_j on all of the condition attributes (i.e., λ) and the distance of x and x_j on certain condition attributes r (i.e., $d_r(x, x_j)$) is minimum, then this condition attribute r can discern the objects x and x_j .

Third, a rule covering system is proposed for the first time in [13]. In this covering system, each rule is induced from the original object, and its covering power can be represented by a circle with a center and a radius (i.e., the reduced object and its corresponding consistence degree).

Definition 2.1 (Rough Classifier): In $FD = (U, R \cup D)$, the set of rules is called a rough classifier, denoted by *Classifier*, if the rules satisfy the following statements: (a) The set of rules covers all of the objects in $FD = (U, R \cup D)$; (b) The set of rules is the minimal set that satisfies (a). Please note that this classifier is not unique because the rule set that satisfies (a) is not unique.

The lower boundary value on discerning objects

On a certain attribute, the similarity degree between two objects, i.e., $r_s(x, x_j)$, is fixed. This similarity $r_s(x, x_j)$ could be the lower boundary value for discerning these two objects. In other words, the object pair can be discerned once we set the threshold larger than this similarity; otherwise, this object pair cannot be discerned by this attribute. The following Theorem

3.1 and 3.2 show this fact.

Theorem 3.1: For $D(x, x_j) = 0$, if $p_{js} = \min\{\beta: 0 < \lambda_\beta - d_{r_s}(x, x_j) \leq \beta\}$, then $\forall \rho \geq p_{js}$, $\max(\lambda_\rho - d_{r_s}(x, x_j), 0) \leq \rho$ always holds.

Proof: $\lambda_\rho = \frac{R_{\beta_{TL}}}{\rho} A(x) = \inf_{D(x,u)=0} \min(1 - R(x, u) + \rho, 1)$; $\lambda_\beta > d_{r_s}(x, x_j) \Rightarrow \lambda_\rho > d_{r_s}(x, x_j)$

There are two cases: 1) if $1 - \max_{D(x,u)=0} R(x, u) + \rho \leq 1$, then $\lambda_\rho = 1 - \max_{D(x,u)=0} R(x, u) + \rho$; and 2) if $1 - \max_{D(x,u)=0} R(x, u) + \rho > 1$, then $\lambda_\rho = 1$.

1) $\rho \geq p_{js}$

and $1 - \max_{D(x,u)=0} R(x, u) + \rho \leq 1$

$1 - \max_{D(x,u)=0} R(x, u) + p_{js} \leq 1 \Rightarrow \lambda_{p_{js}} = 1 -$

$\max_{D(x,u)=0} R(x, u) + p_{js} \Rightarrow$

$1 - \max_{D(x,u)=0} R(x, u) + p_{js} - d_{r_s}(x, x_j) \leq p_{js} \Rightarrow$

$1 - \max_{D(x,u)=0} R(x, u) - d_{r_s}(x, x_j) \leq 0 \Rightarrow 1 -$

$\max_{D(x,u)=0} R(x, u) + \rho - d_{r_s}(x, x_j) \leq \rho \Rightarrow \lambda_\rho -$

$d_{r_s}(x, x_j) \leq \rho$

2) $1 - \max_{D(x,u)=0} R(x, u) + \rho > 1 \Rightarrow \rho > \max_{D(x,u)=0} R(x, u);$

$\rho \geq p_{js} \Rightarrow$

$1 - \max_{D(x,u)=0} R(x, u) + p_{js} - d_{r_s}(x, x_j) \leq p_{js}$ or $1 - d_{r_s}(x, x_j) \leq p_{js} \Rightarrow$

$1 - d_{r_s}(x, x_j) \leq \max_{D(x,u)=0} R(x, u)$ or $1 - d_{r_s}(x, x_j) \leq p_{js} \Rightarrow$

$1 - d_{r_s}(x, x_j) \leq \rho$ because $\rho > \max_{D(x,u)=0} R(x, u)$ and $\rho \geq p_{js}$. \Rightarrow

$\lambda_\rho - d_{r_s}(x, x_j) \leq \rho$ because $\lambda_\rho = 1$ when $1 - \max_{D(x,u)=0} R(x, u) + \rho > 1$.

Above all, $\forall \rho \geq p_{js}$, $\max(\lambda_\rho - d_{r_s}(x, x_j), 0) \leq \rho$ holds. \square

Theorem 3.1 shows that $p_{js} = \min\{\beta : 0 < \lambda_\beta - d_{r_s}(x, x_j)\} \leq \beta$ is the minimum parameter to discern the object pair (x, x_j) by using the condition attribute r . In other words, $p_{js} = \min\{\beta : 0 < \lambda_\beta - d_{r_s}(x, x_j)\} \leq \beta$ is the lower boundary value of the condition attribute r 's discernibility. Because β is the parameter for the noise, p_{js} reflects the robustness of r to discern the object pair (x, x_j) . The larger the value of p_{js} is, the lower the sensitivity of r is.

Theorem 3.2: For

$$D(x, x_j) = 0, \text{ if } p_{js} = \min\{\alpha : \max(\lambda_\alpha - d_{r_s}(x, x_j), 0) \leq \alpha\}, \text{ then}$$

$$p_{js} = \begin{cases} r_s(x, x_j), & r_s(x, x_j) > \max_{D(x,u)=0} R(x, u) \\ 0, & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x, u) \end{cases} \quad (5)$$

Proof:

(1) When $\lambda_\beta > d_{r_s}(x, x_j)$, $p_{js} = \min\{\beta : \lambda_\beta - d_{r_s}(x, x_j) \leq \beta\}$. Because $\lambda_\beta = \frac{R_{\beta_{TL}}}{\beta} A(x) = \inf_{D(x,u)=0} \min(1 - R(x, u) + \beta, 1)$, there are two cases for this definition:

- a) if $1 - \max_{D(x,u)=0} R(x, u) + \beta \leq 1$, then $\lambda_\beta = 1 - \max_{D(x,u)=0} R(x, u) + \beta$;
- and b) if $1 - \max_{D(x,u)=0} R(x, u) + \beta > 1$, then $\lambda_\beta = 1$.

$$\begin{aligned}
 & a) 1 - \max_{D(x,u)=0} R(x,u) + \beta \leq 1 \text{ and } \lambda_\beta - d_{r_s}(x, x_j) \leq \beta \Rightarrow \\
 & 1 - \max_{D(x,u)=0} R(x,u) + \beta - d_{r_s}(x, x_j) \leq \beta \Rightarrow \\
 & 1 - \max_{D(x,u)=0} R(x,u) - 1 + r_s(x, x_j) \leq 0 \Rightarrow \\
 & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x,u) \Rightarrow p_{js} = \min\{\beta: \lambda_\beta - \\
 & d_{r_s}(x, x_j) \leq \beta \text{ when } \lambda_\beta > d_{r_s}(x, x_j)\} = 0 \text{ if } r_s(x, x_j) \leq \\
 & \max_{D(x,u)=0} R(x,u) \qquad \qquad \qquad ; \\
 & b) 1 - \max_{D(x,u)=0} R(x,u) + \beta > 1 \\
 & \text{ and } \lambda_\beta - d_{r_s}(x, x_j) \leq \beta \Rightarrow 1 - d_{r_s}(x, x_j) \leq \beta \text{ and } \beta > \max_{D(x,u)=0} R(x,u) \Rightarrow \\
 & r_s(x, x_j) \leq \beta \text{ and } \max_{D(x,u)=0} R(x,u) < \beta \Rightarrow p_{js} = \\
 & \min\{\beta: \lambda_\beta - d_{r_s}(x, x_j) \leq \beta \text{ when } \lambda_\beta > d_{r_s}(x, x_j)\} = \\
 & r_s(x, x_j) \text{ if } r_s(x, x_j) > \max_{D(x,u)=0} R(x,u) \\
 & (2) \text{ When } \lambda_\beta < d_{r_s}(x, x_j), \text{ then } p_{js} = \min\{\beta: 0 \leq \beta\} = 0. \lambda_\beta < d_{r_s}(x, x_j) \Rightarrow \\
 & 1 - \max_{D(x,u)=0} R(x,u) + \beta < d_{r_s}(x, x_j) \Rightarrow \beta < \max_{D(x,u)=0} R(x,u) - r_s(x, x_j) \\
 & \Rightarrow \max_{D(x,u)=0} R(x,u) \geq r_s(x, x_j).
 \end{aligned}$$

Above all, we have

$$p_{js} = \begin{cases} r_s(x, x_j), & r_s(x, x_j) > \max_{D(x,u)=0} R(x,u) \\ 0, & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x,u) \end{cases} \tag{6}$$

In Theorem 3.2, $\max_{D(x,u)=0} R(x,u)$ is the maximum similarity degree of x from the objects in different classes. If the object pair $r_s(x, x_j) \leq \max_{D(x,u)=0} R(x,u)$, then this object pair can be discerned. If $r_s(x, x_j) > \max_{D(x,u)=0} R(x,u)$, then the object pair cannot be discerned unless the noise tolerance is set to be larger than $r_s(x, x_j)$.

Furthermore, Theorems 3.1 and 3.2 show that if the threshold (the robust degree) α is larger than or equal to $p_{js} = \begin{cases} r_s(x, x_j), & r_s(x, x_j) > \max_{D(x,u)=0} R(x,u) \\ 0, & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x,u) \end{cases}$, then the objects x, x_j can be discerned by attribute r_s ; otherwise, these two objects cannot be discerned. Depending on these discoveries, the algorithm to fast building fuzzy rough classifier is designed as follows.

3 To Find the Base of the Nested Classifier

Algorithm 4.1 is the existing algorithm to build a rule based fuzzy rough classifier. It is easy to see that there are many redundant computations. For example, the computation of the lower approximation requires some running time. As a result, in the following we propose a new algorithm to find a fuzzy rough classifier by using the maximum similarity (i.e., $\max_{D(x,u)=0} R(x,u)$).

Algorithm 4.1. (FRC: Fuzzy Rough Classifier)

Input: $S = (U, R, D)$, $R = \{r_1, \dots, r_k\}$.

Output: *RuleSet*.

Step 1: Compute the similarity degree of R and the lower approximation $R([x]_D)(x)$ for every $x \in U$;

Step 2: For $k = 1, \dots, k$, compute $m_j = \{r: \sigma(1 - r(x, x_j), R([x]_D)(x)) \leq 0\}$ for $D(x, x_j) = 0$; otherwise, $m_j = 0$;

Step 3: Compute one sub-minimal value reduct '*RED*' for every $x \in U$;

Step 4: Compute one reduct rule for every $x \in U$, and add them into '*All_rules*';

Step 5: Do while '*All_rules*' is not empty

5.1) Compute the '*Cover_degree*' of every rule in '*All_rules*'.

5.2) Add the rule '*Rule(x_i)*', which has the maximum '*Cover_degree*' into '*RuleSet*';

5.3) Delete the rules cover by '*Rule(x_i)*' from '*All_rules*';

Step 6: Output '*RuleSet*'

Algorithm 4.2. (MSC: maximum similarity-based classifier)

Input: $S = (U, R, D)$, $VRED = \{r_1, \dots, r_k\}$, $U = \{x_1, \dots, x_n\}$, $x \in U$.

Output: *RuleSet*

Step 1: For $j = 1, \dots, n$, and $s = 1, \dots, k$, compute the similarity degree $r_s(x, x_j)$ and $\max_{D(x,u)=0} R(x, u)$;

Step 2: For $s = 1, \dots, k$, compute

$$d_{js} = \begin{cases} 0, & r_s(x, x_j) > \max_{D(x,u)=0} R(x, u) \\ 1, & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x, u) \end{cases} \text{ for } D(x, x_j) = 0; \text{ others } c_{js} = \emptyset;$$

Step 3: For $j = 1, \dots, n$, Compute one sub-minimal value reduct '*RED_j*' based on c_j ;

Step 4: For $j = 1, \dots, n$, compute $c_j = \min_s (c_{js})$ here

$$c_{js} = \begin{cases} 1, & r_s(x, x_j) > \max_{D(x,u)=0} R(x, u) \\ 0, & r_s(x, x_j) \leq \max_{D(x,u)=0} R(x, u) \end{cases} \text{ for } s \in RED_j;$$

Step 5: Do while any $|c_j|$ is not zero

5.1) Add the rule '*x_i*' which has the maximum $|c_i|$ into '*RuleSet*';

5.2) Update c_j by $c_j = c_j - c_i$;

Step 6: Output '*RuleSet*'

In Algorithm 4.1, d_{j_s} is the discernibility vector. cv_{j_s} is the covering vector. The novelty of Algorithm 4.2 is that it is the first time to use the covering vector to induce the rule. The covering vector avoids computing the covering degree in the loop of the inducing rules. This approach saves a substantial amount of time for the algorithm, overall.

4 Numerical Experiments

In this section, we compare FRC with MSC experimentally. Several datasets from UCI Machine Learning Repository [14] are used to compare the scalability of the algorithms. The detailed information of these datasets is shown in Table 1.

Table 1. The detailed information of several datasets from UCI

	DATASETS	OBJECTS	CONDITION	DATA TYPE	CLASSES
1	Diabetes	769	8	real value	2
2	Iris	150	4	real value	3
3	new_thyroid	215	5	real value	3
4	sat_train	1000	36	real value	5
5	Sonar	208	60	real value	2
6	train_usps	250	256	real value	10
7	waveform_noise	1000	40	real value	3
8	Wdbc	569	30	real value	2
9	Wine	178	13	real value	3
10	Wdbc	198	32	real value	2

The comparison results are summarized in Tables 2. & 3. Table 2 shows that FRC and MSC have the same induction results. It is easy to see from Table 3 that, on average, the execution time of FRC is approximately 1.6 times that of MSC. This finding shows that FRC requires more time to build the classifier than MSC.

Table 2. The induction result of the running time between FRC and MSC

	MSC	FRC	No. of objects
wine	11	11	178
wdbc	21	21	569
wdbc	37	37	198
diabetes	181	181	768
iris	11	11	150
new_thyroid	15	15	215
sonar	25	25	208
train_usps	44	44	500
waveform_noise	111	111	1000
sat_train	42	42	1000

Table 3. The comparison of the running time between FRC and MSC

	FRC V.S. MSC
	(FRC running time/MSC running time)
Wine	1.57
Wdbc	1.37
Wpbc	1.90
Diabetes	1.07
Iris	1.41
New_thyroid	1.39
Sonar	2.03
Train_usps	2.85
Waveform_noise	1.17
Sat_train	1.28
Average	1.61

5 Conclusions

This paper studies the basic notion of lower approximation operators. The authors discover that the lower approximation is closely related to the similarity relation, especially the maximum value of similarity degree. Based on this interesting discovery, a fast algorithm is proposed. By numerical experimental comparison, it is easy to find that the proposed fast algorithm is efficient on building a fuzzy rough classifier.

Acknowledgments. This research work is supported by the Macao Science and Technology Development Fund.

References

1. Beynon, M.J.: An Investigation of β -Reduct Selection within the Variable Precision Rough Sets Model. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 114–122. Springer, Heidelberg (2001)
2. Beynon, M.: Reducts within the variable precision rough sets model: a further investigation. *Eur. J. Oper. Res.* **134**, 592–605 (2001)
3. Cornelis, C., De Cock, M., Radzikowska, A.M.: Vaguely Quantified Rough Sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 87–94. Springer, Heidelberg (2007)
4. Cornelis, C., Martín, G.H., Jensen, R., Ślęzak, D.: Feature Selection with Fuzzy Decision Reducts. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 284–291. Springer, Heidelberg (2008)
5. Cornelis, C., Jensen, R., Martín, G.H., Slezak, D.: Attribute selection with fuzzy decision reducts. *Inf. Sci.* **180**(2), 209–224 (2010)
6. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.* **17**, 191–208 (1990)

7. Dubois, D., Prade, H.: Putting rough sets and fuzzy sets together. In: Slowinski, R. (ed.) *Intelligent Decision support: Handbook of applications and advances of the rough sets theory*, pp. 203–232. Kluwer Academic Publishers (1992)
8. Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough sets. *Fuzzy Sets Syst.* **100**, 327–342 (1998)
9. Pawlak, Z.: Rough sets. *Int. J. Comput. & Inf. Sci.* **11**, 341–356 (1982)
10. Pawlak, Z.: *Rough Sets: theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Dordrecht (1991)
11. Zhu, W.: Topological approaches to covering rough sets. *Inf. Sci.* 177 (6) 1499–1508(2007)
12. Zhu, W., Wang, F.Y.: On Three Types of Covering-Based Rough Sets. *IEEE Trans. Knowl. Data Eng.* **19**(8), 1131–1144 (2007)
13. Eric, C.C., Tsang, S.Z.: Decision Table Reduction in KDD: Fuzzy Rough Based Approach. *T. Rough Sets* **11**, 177–188 (2010)
14. UCI, Machine Learning Repository (2005). <http://www.ics.uci.edu/~mlern/MLRepository.html>
15. Zhao, S.Y., Tsang, E.C.C., Chen, D.G., Wang, X.Z.: Building a Rule-Based Classifier—A Fuzzy-Rough Set Approach. *IEEE Trans. Knowl. Data Eng.* **22**(5), 624–638 (2010)

Improved Learning Algorithms and Applications

Surface Electromyography Time Series Analysis for Regaining the Intuitive Grasping Capability After Thumb Amputation

Chithrangi Kaushalya Kumarasinghe^(✉) and D.K. Withanage

Department of Information Technology, Faculty of Information Technology,
University of Moratuwa, Katubadda, Moratuwa, Sri Lanka
{kaushalyak, dkwithanage}@uom.lk

Abstract. The thumb enables most of the hand's functions such as grasping, gripping and pinching. Therefore the amputation of thumb results many difficulties in object manipulation. In this paper, we present an experimental procedure for manipulating an artificial finger to regain the intuitive grasping capability after the thumb amputation. Here we demonstrate a proportional surface electromyography (s-EMG) classifier, which can be used to obtain three key factors for grasping; the motor command from of the user's nervous system, corresponding angle of rotation and the appropriate torque. The system was tested with both amputated and non-amputated subjects. Based on experiments we offer evidence that, our strategy can be used to intuitively manipulate a prosthetic finger in real time. The system provides a dynamic, smooth and anthropomorphic manipulation of prosthetic fingers under a low training time which is around 3 - 5 seconds with a fast response time around 0.5 seconds.

Keywords: Statistical classifiers · Human-centric systems · Robotics and mechatronics · Man-machine interactions · Surface electromyography (s-EMG) · Regression analysis · Prosthesis · Thumb amputation · Augmented reality

1 Introduction

Physical disabilities obstruct the activities of day to day life. Among them, the amputation of arm or leg is commonly encountered. The three major reasons of amputation are disease, trauma and congenital deformities. Based on the information from the National Center for Health Statistics (NCHS), there are 50,000 amputations every year in USA [4]. Furthermore according to statistics, 92 percent of 102,257 amputations among children and adolescents involved one or more fingers due to accidents that are usually happen in domestic and industrial environments [13].

The thumb enables most of the hand's functions such as grasping, gripping and pinching. Therefore amputation of thumb is really devastating. The support of thumb is extremely important for safely manipulating objects in real world. Figure 1 (a) and (b) demonstrates the safe and efficient object manipulation with the normal hand while (c) and (d) demonstrates the unsafe and inefficient object manipulation with thumb amputated hand.

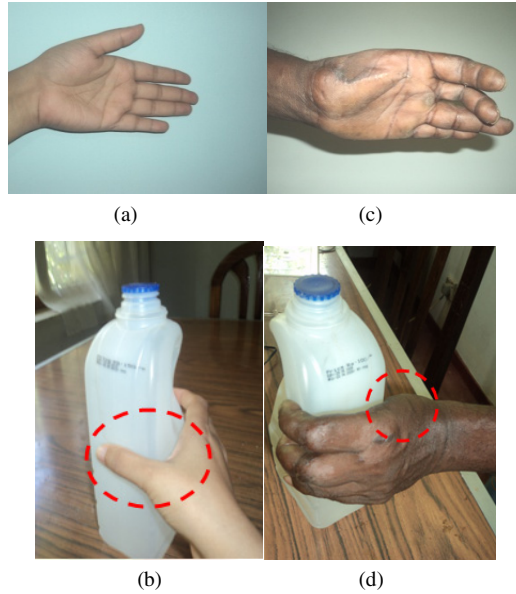


Fig. 1. a) and b) Safe object manipulation with normal hand c) and d) Unsafe object manipulation with thumb-amputated hand

2 Problem Statement

In this paper, we address the issue of regaining the grasping capability after thumb amputation with user's intention, corresponding posture and appropriate torque.

3 Background

Prosthetic limbs improve the quality of life of disabled people. Among them, the myo-electric and body powered prosthetics are functional prosthetics as they support amputees to perform few tasks of the lost limb such as opening and closing of fingers. Most cosmetic artificial limbs are non-functional and they only facilitates the appearance of the lost limb [11].

Surface Electromyography (sEMG) signals which is used in such myo-electric prosthetic devices is a non-invasive medical diagnosis technique to monitor the response of the skeletal muscles for nerve stimulations over the skin surface [11]. The sEMG signals indicate the tension of the skeletal muscles during the flexion and extension. Therefore, several researches have been conducted to derive the angles of rotation in finger joints [12, 6, 8, 9, 15] and as well as the torque of finger tips by using surface EMG signals [5].

The system described in [14] demonstrates an EMG classifier using a graphical hand model. Here, the real time EMG data sets are compared with stored prototypes. The

most matching prototype is considered as the gesture of the hand in real-time manipulation. Further [12] demonstrates the classification of 8 different wrist and finger movements using Support Vector Machine (SVM). A mechanism for a multi grasp myo-electric controller for the transition among 9 different hand gestures using finite state machine was explained in [6]. Here the results are demonstrated using a graphical prosthetic arm.

Even though the performance of myo-electric prosthetic arms have been increased in past few decades in terms of the number of degrees of freedom, number of gestures and response time, most of them was tested with non-amputated subjects. Due to gesture based manipulation they results static and unrealistic movements. Myo-electric prosthetic limbs require a fair amount of active muscles in-order to operate efficiently and accurately. But the characteristics of the surface EMG signal of the monitoring muscle can differ after the amputation as the intuitive controllability of the residual limb differs according to the level of amputation and the activeness of the muscle after the amputation.

To fulfill the above void in prosthesis research, we have extended the existing surface electromyography approach to develop a prosthetic finger with smooth, dynamic and anthropomorphic manipulation to regain the grasping capability after thumb amputation. This paper presents the detailed description of the experimental procedure and the results obtained.

4 Experimental Procedure

The research exhibits three phases. The first step concerns the experimental analysis of the relationship between the angles of finger joints with the s-EMG signal. This was conducted by using non amputees. Then a prototype of a prosthetic arm was developed. Finally, the prototype was manipulated using surface EMG signals according to the derived relationship and tested with both amputees and non-amputees in real time. Figure 2 shows the experimental setup while the basic steps of analyzing the relationship between the angle of rotation and the surface EMG signal are depicted in Figure 3.

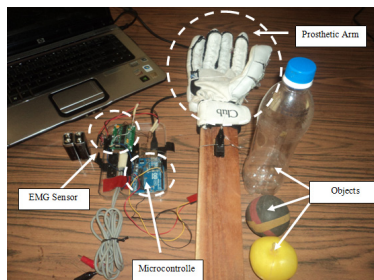


Fig. 2. The experimental setup

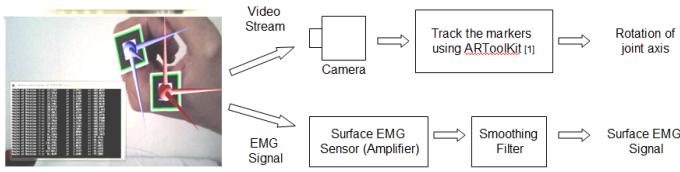


Fig. 3. Basic steps of analyzing the relationship between the angle of rotation and surface EMG signal

4.1 The Relationship Between the Surface EMG Signal and the Angle of Rotation of Finger Joints

In this phase, we analyzed the s-EMG amplitude vs. angle of rotation relationship. The study was conducted by using non amputees. The fore arm muscles were monitored during the opening and closing of fingers while tracing the movements of fingers with augmented reality techniques. Finally we obtained the relationship between the two datasets through regression analysis by using the Matlab Statistics Toolbox [2].

4.2 Surface Electromyography Signal Detection and Pre-processing

The s-EMG signals are acquired by using an amplification chain made up with several high input-impedance amplifiers as described in [7]. The sensor uses three Ag/AgCl (Silver/Silver Chloride) surface EMG electrodes.

The two s-EMG electrodes (mid-muscle electrode and end- muscle electrode) were placed on the Flexor DigitorumProfundus muscle. The third electrode called a reference electrode was placed near the elbow joint (on the posterior side of forearm) as depicted in Figure 4. The voltage difference between the mid-muscle and end-muscle electrodes was measured with respect to the reference electrode.

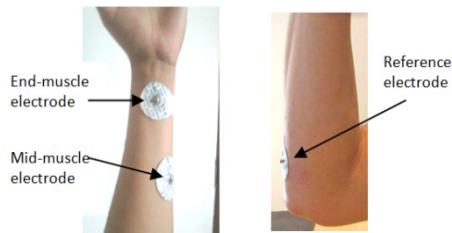


Fig. 4. The placement of surface electromyography electrodes

This analog signal is then converted into a digital signal using an analog to digital converter. The signal is then smoothed by using a moving average filter. Figure 5 (a) shows the change of surface EMG signal with respect to time.

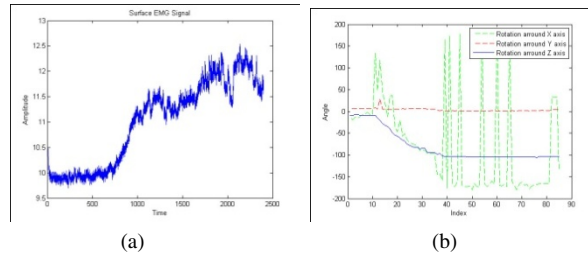


Fig. 5. a) Surface Electromyography signal, b) Rotation of finger joint around x, y and z axis

4.3 Acquisition of Angles of Rotation

We have used the ARToolkit [1], an augmented reality based object tracking technique to track the movements of the fingers using multiple markers. Then the rotational angles are calculated by using the coordinates of the markers. The angle of rotation of the first joint of the index finger is acquired using two markers (marker 1 and marker 2) which were placed on the index finger as shown in figure 3. The transformation of marker 2 with respect to marker 1 was calculated using the ARToolkit as described in [3]. Figure 5 (b) depicts the rotation of the first joint of the index finger around x, y and z axis. During the data acquisition process the subjects were instructed to place their hand in the same position and place the plane of rotation of the measuring finger joint in parallel to the camera lens. With the above two assumptions, we considered the rotation around z axis as the rotation of finger digits to obtain the relationship.

4.4 The Relationship Between the Surface EMG Signal and the Angle of Rotation of Finger Joints

We obtained the relationship between the two datasets using regression analysis. The datasets were fit with a liner regression model as shown in Figure 6. We have noticed the relationship depends on several conditions such as the type of electrodes, location of electrodes, muscle thickness, and background noise.

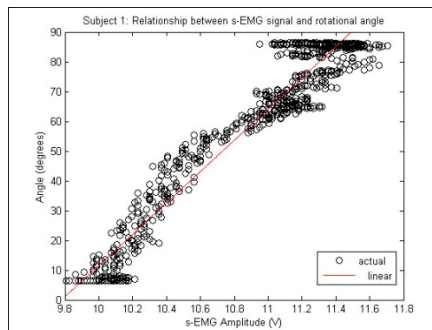


Fig. 6. The relationship between the EMG signal and the angle of rotation

4.5 Real-Time Manipulation of the Prosthetic Arm

After setting up the sensor on user’s forearm, each user was requested to perform one trial finger movement ranging from relaxed state to fully closed state to calibrate the prototype which will take 3 to 5 seconds. During this initialization phase we picked up the minimum and maximum s-EMG samples of the s-EMG time series. The system is then calibrated based on the two values according to a liner function mentioned in previous section. The overall process of real-time manipulation of the artificial arm is depicted in Figure 7.

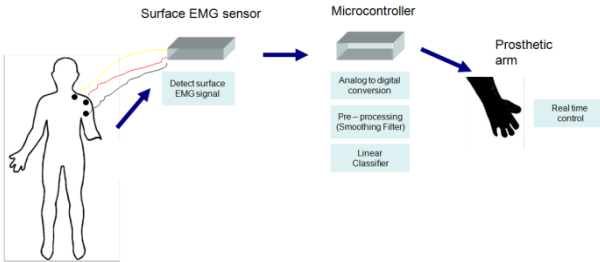


Fig. 7. Steps of calibration and real time manipulation of prosthetic arm

The algorithm for calibrating the prosthetic arm is stated in Figure 8. Here the x_i is the i^{th} sample, A_i is the i^{th} filtered value and n is the window size.

First the Surface EMG signal is parsed through a moving average filter to smooth the signal by removing the noise. Here n is equal to the size of the sliding window and it is equal to 100 where the sampling rate is 500 samples per second. Then the user is advised to perform a single grasping movement that places the remaining fingers of the hand from fully relaxed state to fully contracted state. During this period, the minimum (X_{min}) and maximum (X_{max}) values of the data set will be extracted as the reference points. The minimum value results the thumb opened state while the maximum value results the thumb closed state.

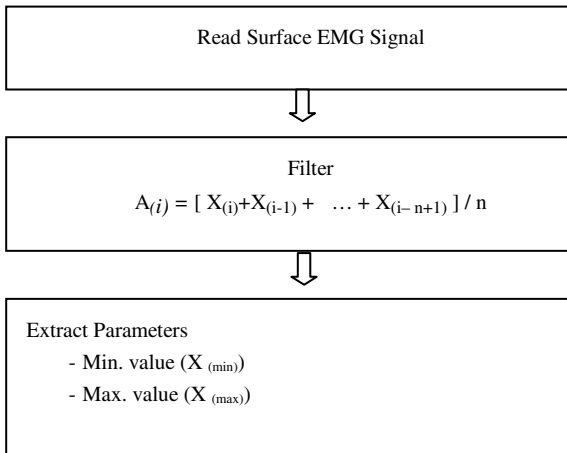


Fig. 8. The calibration process

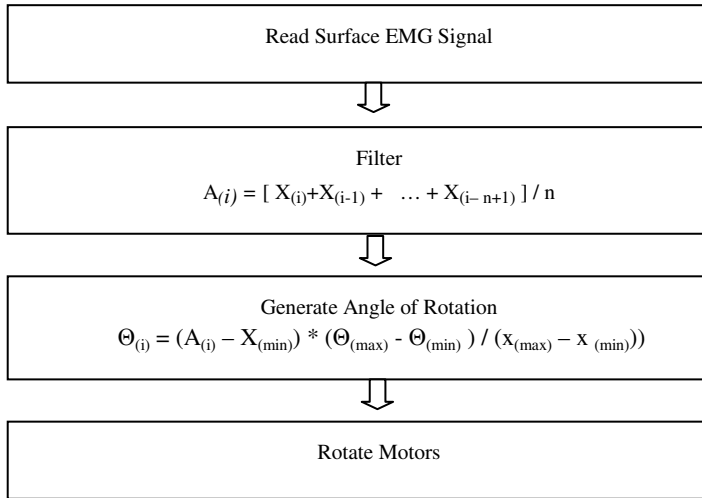


Fig. 9. The process of online manipulation of the prosthetic arm



Fig. 10. Real-time manipulation of prosthetic arm using non-amputees

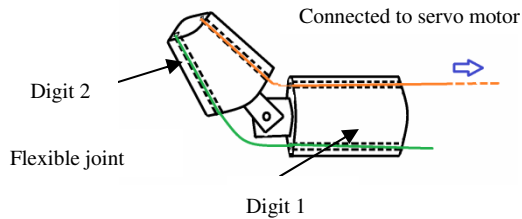


Fig. 11. Prosthetic finger mechanism

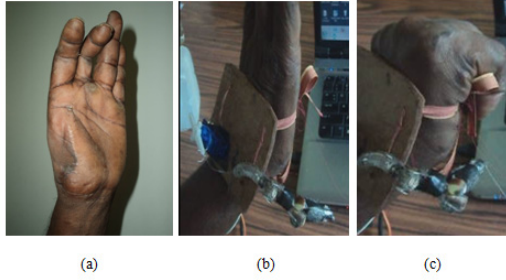


Fig. 12. Real time manipulation of prosthetic finger a) Fore arm of the amputee b) Posture of prosthetic finger when hand opened c) Posture of prosthetic finger when hand closed

These two reference values are used to build the linear classifier. As depicted in Figure 9, each sample of the filtered s-EMG signal is parsed through a linear classifier to obtain the angle of rotation of the prosthetic finger during the real-time manipulation of prosthetic arm.

Figure 10 presents the real-time manipulation of prosthetic prototype using surface EMG. Here, the response time was shorter than .5 second. Table 1 shows the results of the test conducted by using two non amputated subjects while table 2 demonstrates the results of the test conducted with amputated subjects. Several situations have caused the failures in above test.

- Errors done by user during the calibration process
- Displacement of surface EMG electrodes
- Overuse of electrodes

As shown in table 2, we tested the system with a real amputee whose thumb has been removed due to a work related injury happened ten years ago. To demonstrate the functionality we developed a prototype of a prosthetic finger according to the mechanism depicted on figure 11. Further, the figure 12 demonstrates the real-time manipulation of the prosthetic thumb using surface electromyography signals. With experiments we offer evidence that, our system can be used to intuitively control a prosthetic finger while manipulating objects.

Finally, the system was tested to identify the feasibility of grasping objects as shown in Figure 13. At this stage we tested the ability of grasping cylindrical objects such as bottles and rubber balls with non amputees. By using our prototype we were able to successfully grasp those objects.



Fig. 13. Grasping a cylindrical object

Table 1. Results of real-time manipulation of prosthetic arm using non-amputated subjects

Action	Number of succeeded trials	Number of failed trials
Close prosthetic finger	55	5
Open prosthetic finger	57	3
Position the prosthetic finger on a particular angle	5	0
Tight grasping of an cylindrical object	5	0
Fine grasping of an cylindrical object	1	4

Table 2. Results of real-time manipulation of prosthetic thumb using amputated subjects

Action	Number of succeeded trials	Number of failed trials
Close prosthetic thumb according to the movements of other four fingers	7	3
Open prosthetic thumb according to the movements of other four fingers	8	2

5 Discussion

Prosthetic limbs improve the quality of life of disabled people. Several attempts have been made to manipulate prosthetic limbs intuitively by obtaining the nerve stimulations from the user's neural system through implanted electrodes. But it is a costly approach even though it exhibits promising results. However, a lot of amputees do not want to have implanted electrodes inside their body that results in precise motions of their prosthesis. Instead of that, they expect the basic functionalities such as grasping, holding or moving objects.

In this paper, we present our prosthetic system to regain the grasping capability after the thumb amputation. During the initial phase, the system was tested with non-amputees. We were able to obtain high level of accuracy as presented in Table 1. During the latter phase we tested the system with amputees. The user was able to control the prosthetic finger when attempting to grasp the object. While our system only demonstrates the intuitive opening and closing of the prosthetic finger without being capable of grasping an object, our achievement addresses a significant part of the problem. This is an incremental process and we'll be able to mitigate the limitations of the current system by considering the exact physical parameters of the thumb and the torque in our future work.

6 Conclusion

In this paper we described our experimental procedure for regaining the grasping capability after thumb amputation. We have extended the existing proportional surface electromyography technique to identify the proper posture and adequate torque for grasping. The system was tested with both amputees and non-amputees. The results show that this technique facilitates the manipulation of prosthetic finger with the user's intension.

References

1. Artoolkit (2014). <http://www.hitl.washington.edu/artoolkit>
2. Statistics toolbox (2014). <http://www.mathworks.com/help/toolbox/stats/>
3. Tutorial 2: Camera and marker relationships (2014). <http://www.hitl.washington.edu/artoolkit/documentation/tutorialcamera.htm>
4. Panagiotis, K.A., Kostas, J.K.: An emg based robot control scheme robust to time-varying emg signal features. *IEEE Transactions on Information Technology in Biomedicine* **14** (2010)
5. Bida, O., Rancourt, D., Clancy, E.A.: Electromyogram (emg) amplitude estimation and joint torque model performance. In: *Proceedings of the IEEE 31st Annual Bioengineering Conference* (2005)
6. Dalley, S.K., Varol, H.A., Goldfarb, M.: A method for the control of multigrasp myoelectric prosthetic hands. In: *Proceedings of the IEEE 31st Annual Bioengineering Conference*, vol. 20 (2012)
7. Gundanium, Diy muscle sensor / emg circuit for a microcontroller @ONLINE (2013)
8. Kent, B.A., Engeberg, E.D.: Biomimetic myoelectric control of a dexterous artificial hand for prosthetic applications. In: *IEEE Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics* (2011)
9. Khokhar, Z.O., Xiao, Z.G., Menon, C.: Surface emg pattern recognition for real-time control of a wrist exoskeleton. *BioMedical Engineering OnLine* (2010)
10. Lucas, L., DiCicco, M., Matsuoka, Y.: An emg controlled hand exoskeleton for natural pinching. *BioMedical Engineering OnLine* (2004)
11. Micera, S., Carpaneto, J., Raspopovic, S.: Control of hand prostheses using peripheral information. In: *IEEE Reviews in Biomedical Engineering* **3** (2010)
12. Shenoy, P., Miller, K.J., Crawford, B., Rao, R.P.N.: Online electromyographic control of a robotic prosthesis. In: *Proceedings of the IEEE 31st Annual Bioengineering Conference*, vol. 55 (2008)
13. Smith, D.G.: Partial-hand amputations. *Motion*, 17
14. Su1, Y., Wolczowski, A., Fisher, M.H., Bell, G.D., Burn, D., Gao, R.: Towards an emg controlled prosthetic hand using a 3d electromagnetic positioning system. In: *Instrumentation and Measurement Technology Conference* (2005)
15. Tsujimura1, T., Yamamoto1, S., Izumi1, K.: Hand sign classification employing myoelectric signals of forearm

Study on Orthogonal Basis NN-Based Storage Modelling for Lake Hume of Upper Murray River, Australia

Ying Li^{1(✉)}, Yan Li¹, and Xiaofen Wang²

¹ Faculty of Health, Engineering and Science, University of Southern Queensland,
Toowoomba, QLD 4350, Australia

yingli918@gmail.com, yan.li@usq.edu.au

² Department of Sports Engineering and Information Technology,
Wuhan Institute of Physical Education, Wuhan 430000, China

wxiaofen@email.wipe.edu.cn

Abstract. The Murray-Darling Basin is Australia's most iconic and the largest catchment. It is also one of the largest river systems in the world and one of the driest. For managing the sustainable use of the Basin's water, hydrological modelling plays important role. The main models in use are the mathematical represented models which are difficult of containing full relationship between rainfall runoff, flow routing, upstream storage, evaporation and other water losses. Hume Reservoir is the main supply storage and one of the two major headwater storages for the River Murray system. It is crucial in managing flows and securing water supplies along the entire River Murray System, including Adelaide. In this paper, two Orthogonal Basis NN-Based storage models for Hume Reservoir are developed by using flow data from upstream gauge stations. One is only considering flow data from upstream gauge stations. Another is considering both upstream flow data and rainfall. The Neural Network (NN) learning algorithm is based on Ying Li's previous research outcome. The modelling results proved that the approach has high accuracy, good adaptability and extensive applicability.

Keywords: Application · Neural network · Modelling · Orthogonal basis transfer function · Water storage

1 Introduction

The Murray-Darling Basin (the Basin) is Australia's most iconic and the largest catchment. It is also one of the largest river systems in the world and one of the driest. The Basin has four states and one territory (Queensland, New South Wales, Victoria, South Australia and the Australian Capital Territory) [1]. It covers 1 million km². It is ecologically diverse, supporting a wide range of nationally and internationally significant plants, animals and ecosystems including many threatened species. The Basin is Australia's most important agricultural area producing over one-third of Australia's food supply. It yields over 40% of the national agricultural produce and generating around \$15 billion per year for the national economy. It is also home to more than 2

million residents. More than 1.3 million people who live outside the Basin depend on its water resources. The Basin's natural environment plays a vital role in contributing to local production and the national economy [2].

However for the past few years, the health of the Basin is in decline. The ecosystems which rely on the water flowing through the Basin's rivers and tributaries are under considerable pressure, due to unsustainable extraction levels for irrigation and other extractive uses. This problem is likely to become worse as the water availability declines, due to climate change. The Australian Government is committed to restoring the health of our rivers by investing in more efficient water use and delivery, by finding new sources of water, and buying water entitlements from interested sellers to return water to the environment [3].

In 2007, supported by both sides of Federal Parliament, the Water Act was passed to deal with the management of water resource in the Basin in the national interest. The Water Act tasked the Murray-Darling Basin Authority (MDBA) with preparing a Basin Plan to provide integrated management of the Basin water. The Basin Plan is an adaptive framework and will be rolled out over seven years to allow time for the Basin states, communities and the Australian Government to work together to manage the changes required for a healthy working Basin. The Basin Plan limits water use at environmentally sustainable levels by determining long-term average sustainable diversion limits for both surface water and groundwater resource [1]. The proposed Basin Plan again received bipartisan support in the Australian Parliament on 29th November 2012.

For undertaking detailed assessments of environmental water requirements for a range of sites across the Basin, the hydrological modelling has been employed as a fundamental part for developing the Basin Plan, and a daily tool for managing the water supply, water quality and salinity. The models are provided by the Basin states, MDBA and the Australian Commonwealth Scientific and Industrial Research Organization (CSIRO). The main models in use are IQQM, REALM, MSM-Bigmod, WaterCress, and custom-designed models for the Saint George system and the Snowy Mountains Hydro-electric Scheme. Some models have been employed and continue developed for decades. These complex hydrologic modelling tools are mathematical representations of river systems which contain a set of relationships connecting parameters such as river flows, irrigation diversions, water storages, losses (evaporation and overbank flooding), environmental flows, salinity, and ecological state, etc. They simulate the storage, supply and use of water as it is managed for various purposes [4], [5].

Since the complexities and uncertainties of the Basin, the mathematical models are difficult to capture full relationships between rainfall runoff, flow routing, upstream storage, evaporation and other water losses. How to increase the precision of modelling is important for flood management and major water sharing decisions between the environment, irrigation and other extractive uses.

Hume Reservoir (also known as Lake Hume) plays a crucial role in managing flows and securing water supplies along the entire River Murray System, including to Adelaide. The Hume Dam is immediately downstream of the confluence of the River Murray and Mitta Mitta River. The Lake Hume catchment covers less than 1.5% of the total area of the Murray-Darling Basin but contributes 37% of the total inflow to the

Murray River in an average year. These provide immense economic, social and cultural benefits to local and downstream communities along the Murray River. Lake Hume has a catchment area of approximately 15,280km². Approximately 80% of the catchment is forested and approximately 20 percent has been cleared for agriculture [6]. Storage modelling for Lake Hume plays an important role for daily water operation and flood management in the Murray River system.

Ying Li provided the static and dynamic Neural Network (NN) algorithms with orthogonal basis transfer function and their simulation results of water quality (Biological Oxygen Demand) for the Huizhou-Dongan Section of the East River, China [7]. For studying NN-based hydrological modelling and verifying the general applicability of the orthogonal Basis NN-based algorithms, the real observed data is used for water storage modelling for Lake Hume of Upper Murray River, Australia. Two Orthogonal Basis NN-Based models are given in the paper. One considers flow data from upstream gauge stations. Another one considers both upstream flow data and rainfall. The modelling results are given.

2 Orthogonal Basis NN-Based Algorithm

The essential function of NN modelling with one-hidden layer (see Figure 1) is to achieve an unknown input-output (I/O) mapping which can be represented as (1).

$$y = f(x) = \sum_{i=1}^n \theta_i \psi_i(x, \omega_i) \tag{1}$$

Here $\psi_i(x, \omega_i)$ is i th variable basis function which is defined by a transfer function of the hidden layer. n is the neuron number of the hidden layer. ω_i is an adjustable basis parameter.

For N sets sample data $\{x_i, y_{di}, i=1, \dots, N\}$, the vectors in the sample space with the k th input are as (2).

$$\begin{aligned} X_k &= [x_k(1), x_k(2), \dots, x_k(N)]^T \\ F_k &= [f_k(x_k(1)), f_k(x_k(2)), \dots, f_k(x_k(N))]^T \\ \Psi_{ik} &= [\psi_{ik}(x_k(1)), \psi_{ik}(x_k(2)), \dots, \psi_{ik}(x_k(N))]^T \\ \Psi_{xik} &= [x_k(1)\psi_{ik}(x_k(1)), x_k(2)\psi_{ik}(x_k(2)), \\ &\quad \dots, x_k(N)\psi_{ik}(x_k(N))]^T \end{aligned} \tag{2}$$

The norm and inner product are introduced as follows

$$\|F_k\| = \left[\sum_{j=1}^N \omega_{kj} f_k^2(x_k(j)) \right]^{1/2} \tag{3a}$$

$$\langle F_k, \Psi_{ik} \rangle = \sum_{j=1}^N \omega_{kj} f_k(x_k(j)) \psi_{ik}(x_k(j)) \tag{3b}$$

Here ω_{kj} is the weighted coefficient. The optimal objective function is defined as norm square of errors as shown in (4).

$$\begin{aligned} J_k(n) &= \|E_k\|^2 \\ &= \sum_{j=1}^N \omega_{kj} [y_{dj} - f_{kj}(x_k(j))]^2 \\ &= \sum_{j=1}^N \omega_{kj} \left[y_{dj} - \sum_{i=0}^n \theta_{ik} \psi_{ik}(x_k(j)) \right]^2 \end{aligned} \tag{4}$$

Obviously NN learning is searching the optimal square approximation of $f(x)$. The linear transfer functions are employed for input layer and output layer neurons. The orthogonal basis transfer function is used for hidden layer. By Gram-Schmidt orthogonalization method, Zhang proposed a static NN learning algorithm based on orthogonal basis transfer function [8]. For simplifying the NN structure and algorithm, an output f_k is built for each input x_k . The total output with weighted coefficients d_k is defined as follows. Here d_k is calculated by the least square method.

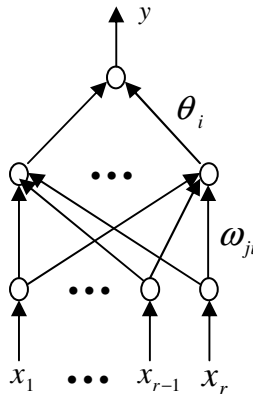


Fig. 1. One hidden layer NN structure

$$f = \sum_{k=1}^r d_k f_k$$

Let $\omega_{kj} = 1$, the recursive formulas are shown in (5), Here $k=1, \dots, r; i=0, 1, \dots, n; n=0, 1, \dots$

$$\psi_{-1k}(x_k) = 0, \psi_{0k}(x_k) = 1, \beta_{0k} = 0, e_0 = y_{d1} \tag{5a}$$

$$\alpha_{(i+1)k} = \frac{\langle \Psi_{x_{ik}}, \Psi_{ik} \rangle}{\langle \Psi_{ik}, \Psi_{ik} \rangle}, \beta_{ik} = \frac{\langle \Psi_{ik}, \Psi_{ik} \rangle}{\langle \Psi_{(i-1)k}, \Psi_{(i-1)k} \rangle} \tag{5b}$$

$$\psi_{(i+1)k}(x_k) = (x_k - \alpha_{(i+1)k})\psi_{ik}(x_k) - \beta_{ik}\psi_{(i-1)k}(x_k)$$

$$\theta_{ik} = \frac{\langle E_{k-1}, \Psi_{ik} \rangle}{\langle \Psi_{ik}, \Psi_{ik} \rangle} \tag{5c}$$

$$f_k(x_k) = \sum_{i=0}^n \theta_{ik} \psi_{ik}(x_k) \tag{5d}$$

$$e_k = e_{k-1} - f_k(x_k), J_k(n) = \|E_k\|^2 \tag{5e}$$

Here $J_k(n)$ in (5) is decreasing which demonstrated by Zhang [8]. Therefore $J_k(n)$ can be calculated to required precision.

For a single output NN system, the optimal objective function defined in (4) is shown as follows

$$J_k(n) = \|e_k\|^2 = \left[y_d - \sum_{i=0}^n \theta_{ik} \psi_{ik}(x_k) \right]^2 \tag{6}$$

The recursive formulas (5b) and (5c) can be modified to the scalar inner product which shown in (7) and (8).

$$\alpha_{(i+1)k} = \frac{\langle x_k \Psi_{ik}, \Psi_{ik} \rangle}{\langle \Psi_{ik}, \Psi_{ik} \rangle}, \beta_{ik} = \frac{\langle \Psi_{ik}, \Psi_{ik} \rangle}{\langle \Psi_{(i-1)k}, \Psi_{(i-1)k} \rangle} \tag{7}$$

$$\theta_{ik} = \frac{\langle e_{k-1}, \Psi_{ik} \rangle}{\langle \Psi_{ik}, \Psi_{ik} \rangle} \tag{8}$$

Now we have the learning algorithm for the orthogonal basis transfer function. Since θ_{ik} is related to Ψ_{ik} , so $\|\theta_{ik} \Psi_{ik}\|^2$ can be a criterion for selecting a transfer function.

3 NN Input Data Selection

All observed data used in this paper are from Live River Data in MDBA’s website [9] and Climate Data Online in the website of Bureau of Meteorology (BOM), Australia [10].

3.1 Murray River Flow Data Selection

Murray River is the boundary between New South Wales and Victoria. The gauge stations located upstream of Lake Hume are shown in Figure 2 [9]. In which following measurement points are used for modelling study:

- 4 -- Jingellic
- 5 -- Hume Dam
- 9 – Tallandoon

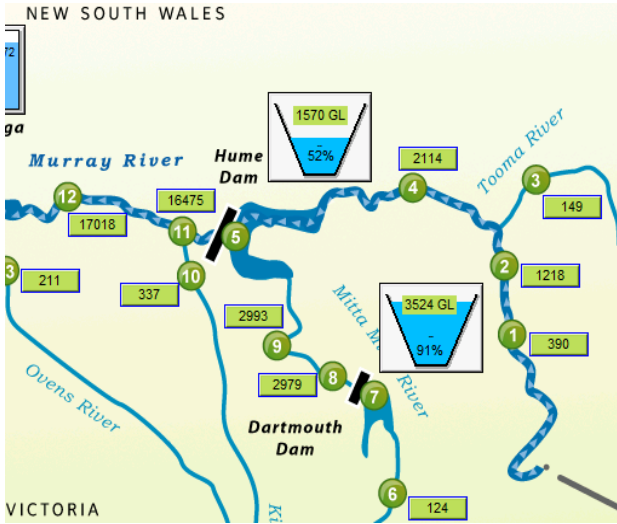


Fig. 2. Murray River gauge stations upstream of Lake Hume

Considering the flow travel time, measurements shown in Table 1 are selected as NN inputs. The 2-Day Mean is the average of current day and next day. Since the data for irrigation and other extractive uses are not publically available, we count the water usage into water losses.

3.2 Rainfall Station Selection

The main rainfall stations upstream of Lake Hume with elevation are shown in Figure 3 [10].

The rainfall station Bethanga is located near Lake Hume but closed in 1970 so not be considered. The rainfall station Tangambalanga NE Dairy is not far from Lake Hume but lower elevation so not be considered. The selected daily rainfall stations for modelling are shown in Table 2, in which the data availability is on 19th Feb 2014 [10].

For simplifying the modelling process and the NN structure, the rainfall measuring points are divided into 3 groups by location and elevation. The mean rainfall of each group is employed. Considering rainfall runoff time and distance, an impact factor (IF) is used for each group value which calculated as (9). Here m is rainfall station numbers in each group.

Table 1. NN input/output with daily data

Input	Explanation
x_1	Lake Hume Previous Day Storage (GL)
x_2	Lake Hume Release (GL/day)
x_3	Murray River Flow at Jingellic (GL/day)
x_4	Previous Day Murray River Flow at Jingellic (GL/day)
x_5	Mitta Mitta River Flow at Tallandoon (GL/day)
x_6	2-Day Mean Water Level at Jingellic (m)
x_7	2-Day Mean Water Level at Tallandoon (m)
Output	Explanation
y	Lake Hume Storage (GL)

Table 2. Selected weather station

GroupNo / NN In	IF	Station No.	Station Name	Distance to Hume Dam (km)	Elevation (m)	First Data	Last Data	Station Status
1 / x_8	1	72023	Hume Reservoir	2.72	184	1922.01	2014.02	open
		72168	Albury (Table Top (Nyrimbla))	11.54	205	2004.01	2012.11	open
		82172	Huon (Barkleys Camp)	19.31	207	2011.08	2014.01	open
		82045	Tallangatta Dcnr	19.36	205	1880.04	2014.02	open
		72050	Wymah (Gwandallen)	21.2	200	1912.02	2014.01	open
2 / x_9	0.85	82047	Tallangatta (Bullioh)	30.53	220	1887.11	2013.12	open
		82102	Bullioh (Hindleton)	30.92	320	1968.11	2014.01	open
		82103	Burrowye Satation	45.79	224	1969.01	2014.01	open
		72144	Tabletop (Table Top (Eastgate))	21.42	273	1966.01	2013.05	closed
		72171	Woomargam Post Office	33	305	2009.07	2014.01	open
3 / x_{10}	0.7	82024	Koetong	41.32	595	1884.04	2014.02	open
		82139	Hunters Hill	46.98	981	1994.06	2014.02	open
		82092	Shelley	47.06	745	1962.12	2014.01	open

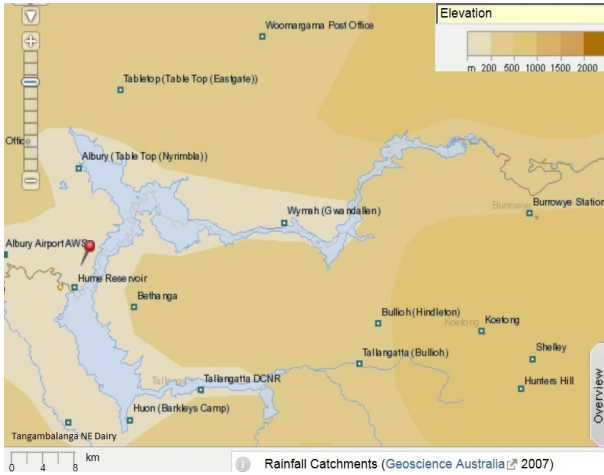


Fig. 3. Main rainfall stations near Lake Hume

$$x(k) = \frac{IF}{m} \left[\sum_{i=1}^m x(k) + \sum_{i=1}^m x(k+1) \right] / 2 \tag{9}$$

4 Modelling Results

The data sets used for modelling are shown in Table 3. The daily data used for modelling are selected from 1st Mar 2009 to 31st Dec 2013 according to the availability. The total number of data is 1767. The data are split into 3 data sets for training, validation and testing. The modelling testing results are shown in Figure 4 to Figure 8. In which the red line is the target and blue line is from the modelling.

The modelling errors are shown in Table 4. The Lake Hume storage modelling errors without considering rainfall are much the same as the one with rainfall. The main reason is that rainfalls in the area are hardly accumulating water flow into the Lake Hume, as the result of absence of infiltration and high evaporation.

Table 3. Modelling data sets

Data Set	Data Period	Data Number	Purpose
Training	01/03/2009 25/01/2012	1061	Update the Weights and Biases
Validation	01/08/2012 12/01/2013	353	Avoid Network Overfitting
Testing	13/01/2013 31/12/2013	353	Check Network Generalizability

Table 4. Modelling errors

Error for Modelling	Mean	Minimum	Maximum
Storage without Rainfall	0.48%	0.0015%	1.77%
Storage with Rainfall	0.48%	0.0012%	1.83%

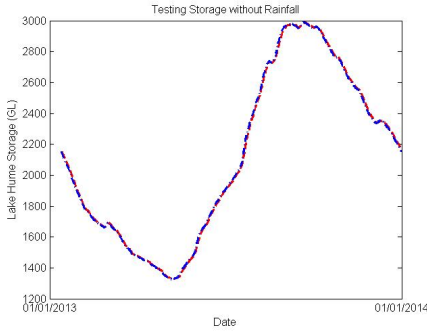


Fig. 4. Lake hume storage without rainfall

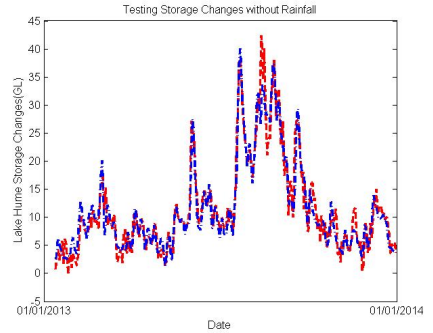


Fig. 5. Lake hume storage changes without rainfall

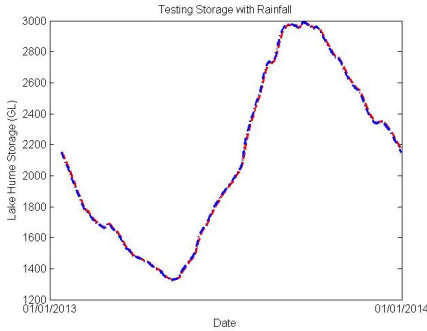


Fig. 6. Lake hume storage with rainfall

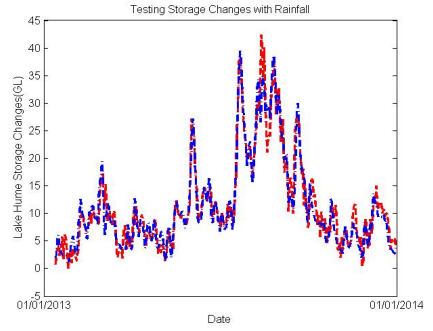


Fig. 7. Lake hume storage changes with rainfall

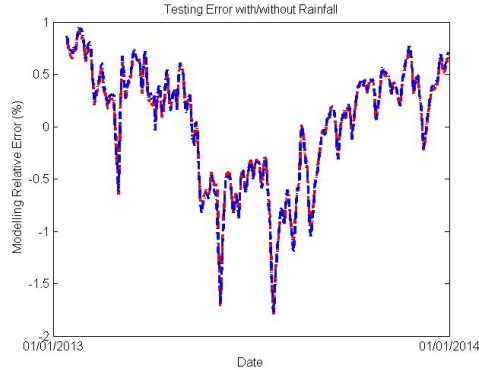


Fig. 8. Modelling errors compare with/without rainfall

5 Conclusions

Lake Hume is the main supply storage and one of the two major headwater storages for the River Murray system, Australia. It is crucial in managing flows and securing water supplies along the entire River Murray System. The hydrological modelling plays important roles for the water management. In this paper, two Orthogonal Basis NN-Based storage models for Hume Reservoir are developed by using flow data from upstream gauge stations. One considers the flow data from upstream gauge stations. Another one considers both upstream the flow data and rainfall. The NN learning algorithm with an orthogonal basis transfer function is based on Ying Li's previous research outcome. The neuron numbers in the hidden layer can be established automatically in the training process. The problem of local extreme value does not exist in the method. The modelling results showed that the approach has high accuracy, good adaptability and may exhibit extensive applicability.

Acknowledgements. Author Ying Li wishes to express her appreciations to MDBA for supporting her to study PhD program in part-time at the University of Southern Queensland, Toowoomba, Australia.

References

1. Commonwealth of Australian: Water Act 2007, Commonwealth of Australian, Canberra (2007)
2. MDBA: Murray-Darling Basin Authority Annual Report 2011-12. MDBA, Canberra (2012)
3. Department of Environment: Restoring the Balance in the Murray-Darling Basin. Department of the Environment, Water Heritage and the Arts, Canberra (2010)
4. MDBA: The proposed environmentally sustainable level of take for surface water of the Murray-Darling Basin: Method and Outcomes. MDBA, Canberra (2011)
5. CSIRO: Uncertainty in river modelling across the Murray-Darling Basin. CSIRO, Australia (2008)

6. Water, G.-M.: Lake Hume Land and On-Water Management Plan. Goulburn-Murray Water (2007)
7. Yang, Y., Li, Y., Zhang, Y.: The Orthogonal Basis NN Based Prediction Modeling for River Water Quality. In: Proceeding of ACC 2001, Arlington, pp. 1611–1615 (June 2001)
8. Zhang, Y.: NN Based Modelling and Control for Nonlinear System, South China University of Technology (1997)
9. MDBA: Live River Data - Murray river upstream of Yarrawonga weir (February 15, 2014). <http://www.mdba.gov.au/river-data/live-river-data>
10. BOM: Climate Data Online (February 15, 2014). <http://www.bom.gov.au/climate/data/?ref=fr>

De Novo Gene Expression Analysis to Assess the Therapeutic and Toxic Effect of Tachyplesin I on Human Glioblastoma Cell Lines

Hongya Zhao¹(✉), Hong Ding², and Gang Jin²

¹ Instrial Center, Shenzhen Polytechnic, Shenzhen 518055, China
hy.zhao@szpt.edu.cn

² School of Applied Chemistry and Biotechnology,
Shenzhen Polytechnic, Shenzhen 518055, China

Abstract. Tachyplesin I (TP-I) is an antimicrobial peptide isolated from the hemocytes of the horseshoe crab. A series of biochemical analysis has been performed to gain insight into the mechanism of its strong antimicrobial and anti-cancer activity. In this study, we employ the microarray technology to identify the co-regulated gene groups of TP-I on human glioma cell lines. The 3 phenotypes of cell lines are treated with the different doses of TP-I including 1-ug/ml, 4-ug/ml and blank groups. As a result, the differentially expressed genes are identified by the paired-comparison of the phenotypes. Considering the consistency within the replicated samples, only the 2572 differential genes are used for the biclustering analysis. Different from the standard clustering, the biclustering algorithms perform clustering along two dimensions of row and column of the data matrix. Detected local patterns may provide clues about the biological processes associated with different physiological states. With the expression data matrix of significant genes across 9 samples, we performs the geometrical biclustering algorithm to find significant co-expressed genes within every phenotype. The further GO analysis with the co-expressed genes are performed to infer the therapeutic and toxic effect of TP-I on human glioma cell lines at the genome level. Some biological processes are of interests. For example, the process related to actin is significantly enriched in Glioblastoma without the treatment with TP-I. Genes defenses virus with the treatment of TP-I. With the increasing dose of TP-I, some toxic effect such as a defensive response to other organism are shown. Our findings provides an alternative choice in the clinical pharmacy for treating glioma with TP-I.

Keywords: Microarray analysis · Biclustering · Gene expression · Tachyplesin I · Gene ontology

1 Introduction

Tachyplesin-I (TP-I) is originally isolated from the acid extracts of hemocytes of the horseshoe crab *Tachypleuriscaris tridentatus* in 1988 [1]. A series of biochemical analysis shows that it can rapidly inhibit the growth of both Gram-negative and Gram-positive

bacteria at extremely low concentrations [1, 2]. It consists of 17 amino acids (KWCFRVCYRGICYRRCR) with a molecular weight of 2,269 and a pI of 9.93. In addition, it contains two disulfide linkages which cause all six of the basic amino acids (R, arginine; K, lysine) to be exposed on its surface. TP-1 is characterized by a unique α -arginine at the C terminal end and a disulfide-stabilized β -sheet conformation [3].

The cationic nature of TP-I allows it to interact with anionic phospholipids presented in the bacterial membrane and thereby disrupts membrane function of bacteria [4, 5]. For example, among these 17 alpha amino acid residues, 4 cysteine residues constitute two disulfide bonds that contribute to the hemolytic ability of TP-I in blood cells [2]. Researches have shown that TP-I is active against both Gram-positive and Gram-negative bacteria [3], fungi [4], viruses [5], and cancer cells [6]. The peptide can also interact with DNA and inhibit the synthesis of macromolecules. Owing to the advantages of TP-I and its relatively small size, this peptide is a promising candidate of a novel alternative antibiotic for animal, the pharmaceutical industry and the food industry.

In recent years, the microarray technology is developed to be a powerful tool for investigation of functional genes and has become a routine tool in many research laboratories [7, 8, 9, 10]. In comparison with the biochemical structure, the mechanism and the hemolytic ability of TP-I, however, the literatures on genomic analysis of TP-I are very limited. The high-throughput gene expression analysis of TP-I is first performed in our lab [11]. Based on previous results, the next microarray experiment is specially designed to assess the genetic expression of therapeutic and toxic effect of TP-I on human glioma cell lines.

Glioblastoma is the most common malignant brain tumor. It is the second leading cause of cancer-related deaths in children for both males and females under age 20 (leukemia is the top leading cause). Recently, microarray-based technologies are widely used to examine gene expression changes in glioblastoma and several related studies have been published in [12, 13]. However, most of researches focus on the identification of significant genes, genetic pathway, and mechanism of glioma only. Our microarray experiment is designed to discover the genetic effect of the new peptide TP-I on glioma cell lines. Our findings provide alternative choice to the clinical pharmacy in treating glioma.

RNAs from human glioma cell lines treated with different doses of TP-I are extracted and being spotted on the microarray chips in our lab. Based on the microarray expression data, we perform the statistical analysis step by step. In addition to the initial identification of significant genes, we also employ the biclustering to detect the groups of genes who work together in glioma with the interference of TP-I.

This paper is organized as follows: After describing our microarray experiment in Section 2.1, the biclustering method is introduced in Section 2.2. Then, the experiment data is analyzed and the results are shown in Section 3. Significant genes detected in microarray analysis are listed in Section 3.1 and the co-expressed gene patterns are recognized in Section 3.2. The corresponding genomic and function analysis of co-expressed genes are also shown in this section. Conclusion and discussion are given in Section 4.

2 Methodology

In this section, the microarray experiment to assess the therapeutic and toxic effect of TP-I on the human glioblastoma cell lines and the corresponding expression data were described in Section 2.1. Then the biclustering methodology to analyze the experimental data was introduced in Section 2.2.

2.1 Microarray Experiment of Human Glioma Cell Lines Treated with TP-I

In our microarray experiment, TP-I was synthesized by Shenzhen Han Yu Pharmaceutical Co., Ltd. (purity, >95.6%) according to the sequence reported by Nakamura et al. (1988) [1]. The RNA samples of the human glioma cell lines were treated with the 0, 1 and 4 $\mu\text{g}/\text{ml}$ TP-I separately. Each labeled sample was replicated to spot on microarray chips for three times. Thus, 9 samples of glioma cell lines are prepared for the RNA extraction.

Total RNA from each sample was quantified by the NanoDrop ND-1000 and RNA integrity was assessed by standard denaturing agarose gel electrophoresis. For microarray analysis, Agilent Array platform was employed. The sample preparation and microarray hybridization were performed based on the manufacturer's standard protocols. Briefly, total RNA from each sample was amplified and transcribed into fluorescent cRNA with using the manufacturer's Agilent's Quick Amp Labeling protocol (version 5.7, Agilent Technologies). The labeled cRNAs were hybridized onto the Whole Human Genome Oligo Microarray (4 x 44K, Agilent Technologies). After having washed the slides, the arrays were scanned by the Agilent Scanner G2505C. Agilent Feature Extraction software (version 11.0.1.1) was used to analyze acquired array images. Quantile normalization and subsequent data processing were performed using the GeneSpring GX v12.0 software package (Agilent Technologies). After quantile normalization of the raw data, genes that at least 9 out of 9 samples have flags in Detected ("All Targets Value") were chosen for further data analysis.

The quality control of RNA integrity and gDNA contamination test in our experiments was shown in Figure 1. The data quality was assessed by the box plot demonstrated in Figure 2. The notations of S0-1, S0-2 and S0-3 were the blank samples without TP-I. And S10-1, S10-2 and S10-3 were the RNA extractions from the cell lines treated with 1 $\mu\text{g}/\text{ml}$ TP-I. Similarly S40-1, S40-2 and S40-3 were the samples treated with 4 $\mu\text{g}/\text{ml}$ TP-I. From Fig. 2, the distribution of the normalized data between and within chips may be hypothesized to be consistent and used for the further analysis.

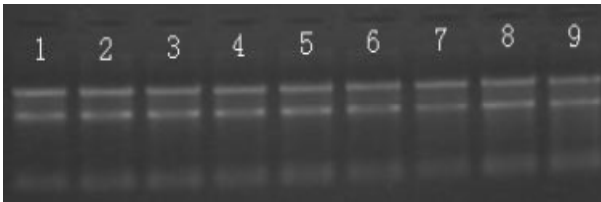


Fig. 1. The purity and integrity of total RNA isolated using NanoDrop-1000 as demonstrated by denaturing agarose gel electrophoresis

In summary, the expression values of 17306 transcript sequences in 9 samples of the human glioblastoma cell lines treated with the different doses of TP-I were recorded. And the hierarchical clustering of expression data in each phenotype was shown in Figure 3. Obviously, the data structure in the three replicated samples of blank and 4 μ g/ml TP-I were very close because they were first merged together. But the data in the sample of S10-1 may be a little distinguishable from S10-2 and S10-3 and it was clustered to the blank sample. The solution of this problem was discussed in Figure 4 of Section 3.1 again.

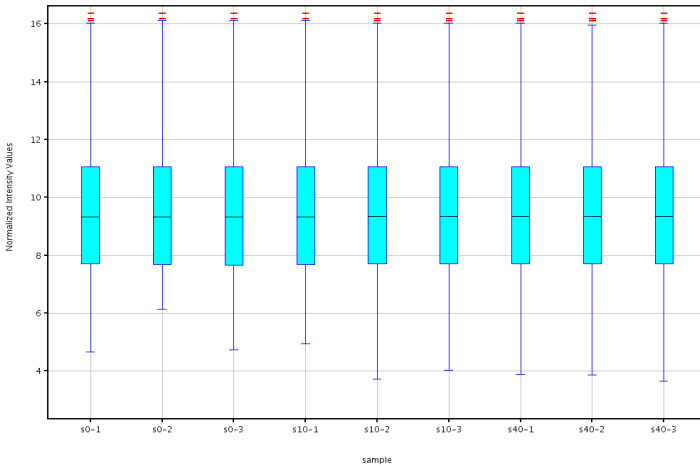


Fig. 2. The boxplot of the normalized expression data of glioblastoma samples treated with 1- and 4- μ g/ml TP-I and blank samples

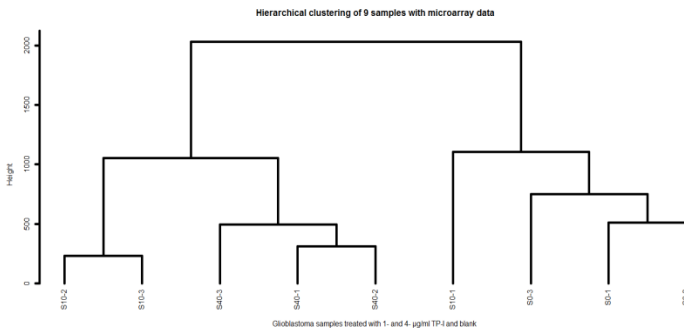


Fig. 3. The hierarchical clustering of 9 glioblastoma samples treated with 1- and 4- μ g/ml TP-I and blank

2.2 Biclustering Analysis

Biclustering analysis is a useful methodology to discover the local coherent patterns hidden in a data matrix. Unlike the traditional clustering procedure, which searches

for groups of coherent patterns using the entire feature set, biclustering performs simultaneous pattern classification in both row and column directions in a data matrix. The technique has found useful applications in many fields but notably in bioinformatics [14, 15, 16, 17]. For a review of the biclustering technology, the reader is directed to [18, 19, 20].

Denote the gene expression matrix $D_{N \times n}$ with N genes and n experimental conditions. Traditional clustering attempts to group objects (genes or conditions) into different categories to uncover any hidden patterns. For example in the hierarchical clustering in Figure 3, all of the genes on the chip are considered as the features to calculate the distance and the samples are classified only along the columns. Comparatively, biclustering performs clustering in the gene and condition dimensions simultaneously. In a genome, a bicluster is regarded as being a set of genes that exhibit similar biological functions under a subset of experiment conditions and vice versa [18-20].

Denoting the index of D as $G = \{g_1, \dots, g_N\}$ and $C = \{c_1, \dots, c_n\}$, we have $D = (G, C) \in \mathbb{R}_{|G| \times |C|}$. Thus, during data analysis a bicluster $B = (X, Y)$ appears as a sub-matrix of D with some specific patterns, where $X = \{N_1, \dots, N_x\} \subseteq G$ and $Y = \{n_1, \dots, n_y\} \subseteq C$ are a separate subset of G and C . Five coherent types of biclusters were reviewed by Madeira and Oliveira, including constant, constant rows, constant columns, additive and multiplicative ones, corresponding to different biological phenomena [18]. Various biclustering algorithms are proposed. Most of these algorithms employ data mining techniques to search for the best possible sub-matrices. The general strategy in all these algorithms can be described as permuting rows and/or columns of the data matrix in a number of ways such that an appropriate merit function is improved by the action.

In contrast to the existing permutation-based approach, a novel geometric perspective for the biclustering problem is proposed in [20, 21, 22, 23, 24]. In this viewpoint, sub-matrices become points in the high dimensional data space. Instead of searching for coherent pattern in data matrix by permutation processes, the biclustering is transformed to the detection of geometric structures formed by spatial arrangement of these data points. This perspective first provides a unified formulation for extracting different types of biclusters simultaneously. Furthermore, the geometric view makes it possible to perform biclustering with the generic line or plane finding algorithms. Some geometric biclustering algorithms such as GBC and RGBC are proposed in [22, 23, 24]. The RGBC algorithm is used in our data analysis in the paper.

3 Results

3.1 Identification of Differentially Expressed Genes

As mentioned in Section 2. 1, we obtained the microarray data matrix $D_{N \times n} = (d_{ij})$, $i = 1, \dots, 17306$; $j = 1, \dots, 9$. And the differentially expressed genes can be identified with statistical significance between any two of 1-, 4- $\mu\text{g/ml}$ TP-I and blank samples. Considering the homogeneous distribution of the normalized data, the direct 1.5-fold change with $p < 0.05$ in t-test was employed to identify the significant genes. The traditional t-test method can only compare between the treatment and control

samples. But there are two treatment groups in our microarray experiment. So the three hypotheses of t-test between any two groups were made.

We first compared the samples treated with 1 $\mu\text{g/ml}$ TP-I and the blank. The 653 differentially expressed genes were identified, in which there were 482 up-regulated genes and 171 down-regulated. Similarly, the 2352 significant genes were identified by comparing the expression data between 4 $\mu\text{g/ml}$ TP-I treatment group and the blank. And the 1304 genes were up-regulated and 1048 were down-regulated. Furthermore we also detected the differentially expressed genes between the two groups respectively treated with 1 and 4 $\mu\text{g/ml}$ TP-I. 278 up-regulated and 325 down regulated genes were selected.

Obviously there were some overlapped genes among the significant genes mentioned in the previous part. We filtered 2572 differentially expressed genes in the complete comparisons as the candidates for the further biclustering analysis. The heatmap and hierarchical clustering of the 2572 genes across 9 samples are demonstrated in Figure 4. Different from the hierarchical clustering with the expression data of all genes in Fig. 3, the three replicated samples with the same treatment were first clustered in Fig. 4. That is, S0-1, S0-2 and S03 were first merged into one group; S10-1, S10-2 and S10-3 were in one group and the same to S40-1, S40-2 and S40-3.

In fact, the number of the normal genes that are not altered with the treatment is much larger than that of the significant. Considering the reasonable change from Fig. 3 to 4, it seemed that a large amount of the expression values of the normal genes distorted the expression pattern of the significant genes. So it is meaningful to only employ the expression data of the significant genes for the further analysis.

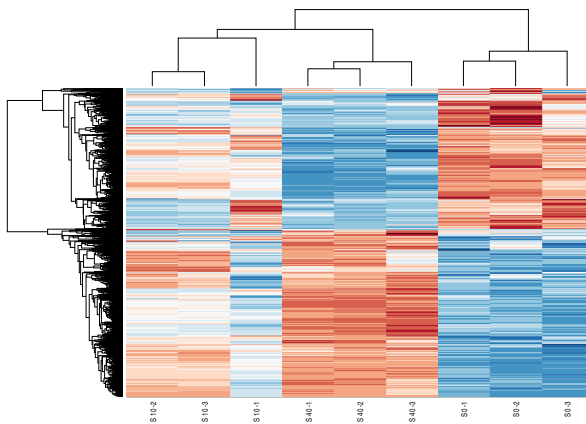


Fig. 4. The heatmap and hierarchical clustering of 2572 differentially expressed genes across 9 glioblastoma samples treated with 1- and 4- $\mu\text{g/ml}$ TP-I and blank

3.2 Biclustering Analysis of Co-expressed Gene Patterns and GO

As mentioned in Section 2.2, the geometrical biclustering algorithm, RGBC, was used to detect the local patterns along the two dimensions of genes and conditions of our

data matrix of the differentially expressed genes. The Matlab codes of RGBC algorithm can be downloaded from the web link <http://www.hy8.com/bioinformatics.htm>. The resulted geometrical biclusters can be filtered if the minimum number of genes is less than 10.

It is well known that co-expressed genes, not one gene, are involved in the biological function together. So we investigate the biological meaning of the significant genes belonging to one bicluster with the same phenotypes with Gene Ontology (GO) framework [25].

As demonstrated in the hierarchical clustering of Fig. 4, the three glioblastoma samples (S40-1, S40-2 and S40-3) treated with 4ug/ml TP-I were first grouped. In fact, the number of the biclusters including the 3 samples was also large. So we selected one with the largest number of genes for the following GO analysis. And there was no bicluster detected with all of the 3 glioblastoma samples treated with 1ug/ml (S10-1, S10-2 and S10-3). It seemed that the distribution of expression data in the three samples was not uniform, even only with the significant genes. Then considering the high similarity of S10-2 and S10-3 in the hierarchical clustering of Fig. 3 and 4, the largest bicluster with the two samples are selected for GO analysis. Some biclusters with the columns of S0-1, S0-2, S0-3 and S10-1 can be detected. As a result, we also considered one largest bicluster with the four samples for GO analysis.

The results of GO analysis for the three phenotypes were summarized in Table 1. The first column is the indexes of the largest biclusters for the three phenotypes separately. The samples of the first bicluster are S0-1, S0-2, S0-3 and S10-1. Although S10-1 was merged into the bicluster, we still thought that the corresponding biological process was followed by the significant genes in the human glioblastoma. The second one includes S10-2 and S10-3. The GO terms may explain the genetic co-expressed patterns with the treatment of 1 ml/ug TP-I. The last one is S40-1, S40-2 and S40-3 and can be considered as the significant GO features with the treatment of 4ml/ug TP-I.

The table has the following organization: Representative GO terms and its biological meaning for genes are listed in the first and second column. For each GO term, we report four terms of enrichment in the last column, including the total number of genes (N), the number of genes annotated to the GO term (B), the number of genes from our assayed gene annotated to it (n), the number of genes in the intersection (b). The corresponding p-values of the GO terms enriched were computed according to the following hyper-geometric distribution and the values were listed in the fourth column of Table 1.

$$p\text{-value} = \sum_{j=b}^B \frac{\binom{n}{j} \binom{N-n}{B-j}}{\binom{N}{B}} \tag{1}$$

According to Table 1, the GO features of glioblastoma in Bicluster 1 are related to regulation of localization, developmental process, and actin. Indeed, the progress of glioblastoma is significantly dependent on the biological process of actin, which is consistent with the glioma literatures [12,13].

Table 1. The significant biological processes of co-regulated genes on glioblastoma treated with TP-I

Bi-clusters	GO term	Description	P-value
1	GO:0032879	regulation of localization	8.85E-10
	GO:0032502	developmental process	1.01E-08
	GO:0030036	actin cytoskeleton organization	8.42E-08
	GO:0030029	actin filament-based process	9.84E-08
2	GO:0032879	regulation of localization	3.41E-11
	GO:0051128	regulation of cellular component organization	6.37E-09
	GO:0032502	developmental process	3.83E-08
	GO:0042127	regulation of cell proliferation	7.25E-08
3	GO:0051607	defense response to virus	2.02E-24
	GO:0098542	defense response to other organism	8.34E-24
	GO:0071357	cellular response to type I interferon	2.00E-23
	GO:0060337	type I interferon signaling pathway	2.00E-23
	GO:0051707	response to other organism	2.69E-23
	GO:0034340	response to type I interferon	4.11E-23
	GO:0009615	response to virus	2.37E-21
	GO:0043207	response to external biotic stimulus	2.61E-21
	GO:0006952	defense response	6.15E-21

Then the genetic function is altered with the treatment of 1ug/ml TP-I in Bicluster 2. By comparing the GO term in Bicluster 1 and 2, we found that the processes of regulation of localization and developmental process are the same. However, the most importance in the function is defense response to virus. In other words, TP-I just shows the therapeutic effect to glioblastoma. And it is also valuable that TP-I may take effect on the regulations of cellular component organization and cell proliferation to intervene glioma.

With the increasing dose of TP-I from 1ug/ml to 4 ug/ml , the biological processes of defense response and response to virus, other organism and type I interferon become significant in the third bicluster. As mentioned in the previous part, the three samples treated with 4ug/ml TP-I, S40-1, S40-2 and S40-3, were always prone to be merged in our biclustering analysis. And the significant genes in the bicluster are also enriched with the smallest p-values ($<10^{-21}$). Different from the significant inconsistency of the expression data with the treatment of 1 ug/ml, it seemed that the biological processes are highly uniform in the samples treated with 4 ug/ml TP-I. We may conclude that the biological regulation of the co-expressed genes become more significant with the high concentration of TP-I in glioblastoma cell lines. Certainly the toxic effect of TP-I should be considered at this stage. The response and defense response to other organism are also shown.

4 Conclusion

As a novel antimicrobial peptide, Tachyplesin I (TP-I) shows some advantages in its biochemical structure, mechanism and hemolytic ability. This peptide is considered to be a promising candidate of a novel alternative antibiotic in the animal, pharmaceutical industry and food industry. We design the novel microarray experiment to identify the co-regulated genes on human glioma cell lines treated with TP-I. Glioblastoma is the most common malignant brain tumor. It is the second leading cause of cancer-related deaths in children under age 20.

The 3 phenotypes of cell lines are treated with different doses of TP-I including 1-ug/ml, 4-ug/ml and blank groups. The De Novo microarray analysis is performed with our microarray data. The 2572 differentially expressed genes are identified by the paired-comparison of the phenotypes. The significant modification of S10-1 is detected in the hierarchical clustering of Figures 3 and 4. We will perform more works in future to explore it.

Considering the co-regulation in genetic mechanism, we use the biclustering algorithm to find the significant co-expressed genes within every phenotype. The further GO analysis with the co-expressed-genes are performed to infer the therapeutic and toxic effect of TP-I on the human glioma cell lines at genome level. Some biological processes are of interests. For example, the process related to actin is significantly enriched in Glioblastoma without treatment with TP-I. Moreover, genes defense virus with the treatment of TP-I. With the increasing dose of TP-I, some toxic effects, such as the response and defensive response to other organism, are also shown. To elucidate the genetic function of TP-I related to the concentration, more microarray experiments are needed.

Acknowledgments. This study was supported by National Natural Science Funds of China (31100958 and 31272474), GDNSF (10151805501000007 and S2011020005160).

References

1. Nakamura, T., Furunaka, H., Miyata, T., et al.: Tachyplesin, a class of antimicrobial peptide from the hemocytes of the horseshoe crab (*Tachypleustridentatus*), isolation and chemical structure. *Biological Chemistry* **263**, 16709–16713 (1988)
2. Ramamoorthy, S., Thennarasu, A., Tan, K., et al.: Deletion of all cysteines in tachyplesin I abolishes hemolytic activity and retains antimicrobial activity and lipopolysaccharide selective binding. *Biochemistry* **45**, 6529–6540 (2006)
3. Masuda, K., Ohta, M., Ito, M., et al.: Bactericidal action of tachyplesin I against oral streptococci. *Oral Microbiology and Immunology* **9**(2), 77–80 (1994)
4. Ouyang, G.L., Li, Q.F., Peng, X.X., et al.: Effects of tachyplesin on proliferation and differentiation of human hepatocellular carcinoma SMMC-7721 cells. *World Journal of Gastroenterology* **8**(6), 1053–1058 (2002)
5. Gao, Y., Zhao, H., Feng, X., et al.: Expression of Recombinant Human Lysozyme-tachyplesin I (hLYZ-TP I) in *PichiaPastoris* and Analysis of Antibacterial Activity. *Bio-med. Environ. Sci.* **26**(4), 319–322 (2013)
6. Teixeira, V., Feio, M., Bastos, M.: Role of lipids in the interaction of antimicrobial peptides with membranes. *Prog. Lipid Res.* **51**, 149–177 (2012)
7. Chapman, D., Papaioannou, V.: Three neural tubes in mouse embryos with mutations in the T-box gene *Tbx6*. *Nature* **391**(12), 695–697 (1998)

8. Nikaido, M., Kawakami, A., Sawada, A., et al.: Tbx24, encoding a T-box protein, is mutated in the zebrafish somite-segmentation mutant fused somites. *Nature Genetics* **31**(2), 195–199 (2002)
9. Kushibiki, T., Kamiya, M., Aizawa, T., et al.: Interaction between tachyplesin I, peptide derived from horseshoe crab, and lipopolysaccharide. *Biochim. Biophys. Acta* **1844**(3), 527–534 (2014)
10. Qiu, D., Liu, X., Wang, J., Su, Y.: Remove from marked Records Artificial synthesis of TAT PTD-tachyplesin fusion gene by overlap extension PCR. *J. Agricultural Biotechnology* **2**(3), 1–4 (2013)
11. Zhao, H., Dai, J., Jin, G.: Genomic and functional analysis of the toxic effect of tachyplesin I on the embryonic development of zebrafish accepted by *Computational and Mathematical Methods in Medicine*
12. van den Boom, J., Wolter, M., Kuick, R., et al.: Characterization of Gene Expression Profiles Associated with Glioma Progression Using Oligonucleotide-Based Microarray Analysis and Real-Time Reverse Transcription-Polymerase Chain Reaction. *The American Journal of Pathology* **163**(3), 1033–1043 (2003)
13. Magnus, N., Gerges, N., Jabado, N., Rak, J.: Coagulation-related gene expression profile in glioblastoma is defined by molecular disease subtype. *Journal of Thrombosis and Haemostasis* **11**(6), 1197–1200 (2013)
14. Wang, L.P., Chu, F., Xie, W.: Accurate cancer classification using expressions of very few genes. *IEEE-ACM Transactions on Bioinformatics and Computational Biology* **4**(1), 40–53 (2007)
15. Mitra, S., Hayashi, Y.: Bioinformatics with soft computing. *IEEE Transactions on Systems, Man and Cybernetics, Part C* **36**, 616–635 (2006)
16. Tseng, V., Kao, C.: Efficiently mining gene expression data via a novel parameterless clustering method. *IEEE-ACM Transactions on Computational Biology and Bioinformatics* **2**, 355–365 (2005)
17. Cheng, Y., Church, G.M.: Biclustering of Gene-Expression Data. In: *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB 2000)*, pp. 93–103 (2000)
18. Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. Computational Biology and Bioinformatics* **1**(1), 24–45 (2004)
19. Tanay, A., Sharan, R., Shamir, R.: Biclustering Algorithms: A Survey. In: Aluru, S. (ed.) *Handbook of Computational Molecular Biology*. Chapman & Hall (2005)
20. Zhao, H., Liew, A.W.-C., Wang, D.Z., Yan, H.: Biclustering Analysis for Pattern Discovery: Current Techniques. *Comparative Studies and Applications Current Bioinformatics* **7**(1), 43–55 (2012)
21. Gan, X., Liew, A.W.C., Yan, H.: Discovering Biclusters in Gene Expression Data Based on High-Dimensional Linear Geometries. *BMC Bioinformatics* **9**(209) (2008)
22. Zhao, H., Liew, A.W.-C., Yan, H.: A New Strategy of Geometrical Biclustering for Microarray Data Analysis. In: *APBC*, pp. 47–56 (2007)
23. Zhao, H., Liew, A.W.-C., Xie, X., Yan, H.: A new geometric biclustering algorithm based on the Hough transform for analysis of large-scale microarray data. *Journal of Theoretical Biology* **251**, 264–274
24. Zhao, H., Chan, K.-L., Cheng, L.-M., Yan, H.: A probabilistic relaxation labeling framework for reducing the noise effect in geometric biclustering of gene expression data. *Pattern Recognition* **42**(11), 2578–2588 (2009)
25. Ashburner, M., et al.: Gene Ontology: tool for the unification of biology. *Nature Genetics* **25**, 25–29 (2000)

An Improved Reject on Negative Impact Defense

Hongjiang Li and Patrick P.K. Chan^(✉)

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

lhj3lhj@gmail.com, patrickchan@ieee.org

Abstract. Causative attack in which the training samples have been attacked in order to mislead the learning of a classifier is a common scenario in adversarial learning. One of the countermeasures is called the data sanitization which removes suspect attack or noisy samples before training. The data sanitization can be categorized into classifier-independent and classifier-dependent methods. Classifier-independent methods measure the characteristics of the samples while classifiers are trained in classifier-dependent methods. Although the accuracy of classifier-dependent methods is higher, they are time-consuming in comparison with classifier-independent methods. This paper proposes a data sanitization method using both classifier-dependent and classifier-independent information. Not only one sample but a set of similar samples identified by the relative neighborhood graph are considered in Reject on Negative Impact method. The experimental results suggest that the performance of the proposed method is similar to the RONI but with less time complexity.

Keywords: Adversarial classification · RONI · Data sanitization

1 Introduction

Machine learning has been increasingly applied in many security applications [9] due to its promising performance. However, an attacker who intentionally misleads the system by manipulating the samples may present in the security applications [21, 23]. For example, the spam message is camouflaged by adding good words, which appear frequently in legitimate but not in malicious messages, to evade the spam filtering [1]. As more and more attacks we have faced with, the problem of it should not be ignored. The adversarial attack may significantly affect the performance of the system.

The causative attack [2, 10, 18, 22], also named as the poison attack [7], is one of the well-known adversarial attacks. The causative attack aims to mislead the learning process of a classifier by modifying the samples in the training set. Some robust learning algorithms [12, 13, 19] have been proposed to reduce the influence of the attack samples in the training process. The robust learning minimizes not only the empirical risk of a classifier but also the risk of treating a sample is untainted in training. Another defense method against the causative attack is the data sanitization [4]. The suspect samples are identified from the training set according to some criteria and removed before learning. The data sanitization methods can be categorized into two

types according to the classifier dependence. The classifier-independent methods select the attack or noise samples based on the characteristics of the dataset. Most of these methods make use of the relation of distance between instances. For example, in 1NN method [14], an instance will be removed if its label is different from the label of the nearest one. On the other hand, the information provided by a classifier is applied to selecting suspect samples. Reject on Negative Impact (RONI) defense [4] measures the negative impact of a samples to the model which is defined as the difference between the classification accuracies of a classifier trained with / without using that sample. If the classifier trained without a sample is more accurate than the one with the sample, the sample is removed. The advantage of classifier-dependent methods is more accurate generally in comparison with the independent ones. However, the time complexity of dependent methods is much less as no classifier training is involved.

In this paper, we propose a data sanitization method considering the information provided by the structure of the dataset and also the classifier. The time complexity of RONI is relatively high as two classifiers are built for each sample. As a result, the proposed method considers a cluster of samples chosen by relative neighborhood graph [5]. The experimental results show that the running time is reduced by the proposed method and its accuracy is similar to RONI.

The rest of the paper is structured as follows. Section 2 introduces the related works briefly. The proposed hybrid method is described in Section 3. The experimental results are reported in Section 4, and finally, the conclusion is given in Section 5.

2 Background

In this section, the various causative attacks are introduced firstly. Then, the existing data sanitization methods are discussed.

2.1 Causative Attacks

The objective of the causative attack is to maximize the error of a classifier on all samples [11] or the malicious samples [20] with the minimal modification on the training set. The modification can be occurred in feature values [8] or class label [11]. The attack strength is limited by the number of attack samples [4] and also the modification on each sample [4].

Adversarial label flips attack is a common type of causative attack [11]. The adversary poisons the training set through flipping labels. Examples are the nearest-first and the furthest-first attack [11]. The labels of the samples closest (furthest) to the surrogate decision function are flipped in the nearest-first (furthest-first) attack. Both methods mislead the classifier efficiently.

2.2 Data Sanitization

Data sanitization, which removes suspect samples before training is one of the counterattack methods to causative attacks. The methods are categorized into two types

according to whether their selection criteria rely on the information of a classifier. For example, CutEdge method [15, 16], which is classifier-independent method, removes a sample if its label is different from the one of its connected samples according to the relative neighborhood graph.

Differently, classifier-dependent methods rely on the information provided by classifiers trained using the dataset. RONI [4] inspects every instance individually. For each sample, two classifiers are trained (without) using the sample and also a subset randomly selected from the dataset. The sample is removed if the accuracy of the classifier trained by using the sample is lower than the one without using the sample. Another example is micro-models [3]. The training data is divided into multiple subsets randomly. A classifier, namely a multiple model, is trained by using a subset. If most of the classifiers classify a sample wrongly, it will be sanitized from training set.

3 Proposed Data Sanitization Method

It has been shown that RONI has promising performance [4, 6]. However, as two classifiers are trained for each sample, similar to other classifier-dependent methods, the time complexity of RONI is relatively high in comparison with classifier-independent methods. In this section, we introduce a data sanitization which simplifies the calculation of RONI by considering a set of similar samples grouped according to the relative neighborhood graph. Moreover, the performance of the two classifiers of a sample is the same sometimes in RONI since the impact of the sample to the classifier may not be significant. By considering a set of samples, the performance of our proposed method may increase.

Algorithm 1. Proposed Data Sanitization Method

Input: a training set D

Output: the suspected set D^s

1. $D^s = \text{null}$
 2. $C = \text{Cluster } D \text{ by the relative neighborhood graph}$
 3. **for** $i = 1, 2, \dots, |C|$
 4. Random D^T and D^S as training and validation set from $D - C_i$
 5. $F_1 = \text{a SVM trained by using } D^T$
 6. $F_2 = \text{a SVM trained by using } \text{union}(C_i, D^T)$
 7. **if** F_1 is more than F_2 , **then** $D^s = \text{union}(D^s, C_i)$
 8. **end for**
 9. **return** D^s
-

Give a dataset D containing n number of samples, i.e. $D = \{\mathbf{x}_i, y_i\}$, where $i = 1 \dots n$, \mathbf{x} is the feature vector and y is the class label. D is then clustered according to the relative neighborhood graph [17], i.e., sample a and b are connected if there is no sample c exists which the distance between (a, c) or (a, c) is smaller than (a, b) . All samples in a

cluster are considered as a group and will be removed if they are not useful for improving the performance of the classifier is removed. The detail algorithm is shown in algorithm 1.

4 Experiment

The proposed data sanitization is compared experimentally with RONI using TAO_grid (tao), Pima Indians Diabetes (pim), Statlog/Heart (h-s) and Connectionist Bench (Sonar, Mines vs. Rocks) (sonar) datasets from UCI Machine Learning Depository [24], shown in Table 1. The dataset is spitted randomly into training (50%) and testing set (50%). The nearest-first and furthest-first attacks are injected to the training set. The attack strengths are 5%, 10%, 15%, 20% and 25% of the total samples. The proposed method and RONI are applied to sanitize the training set before the training. SVM with the Gaussian kernel is used and its parameters (i.e., $g = 0.5$) are selected by 5 fold cross-validation. Section 4.1 reports the performance of the data sanitization evaluated by using the classification error on the testing set. Moreover, the running time is also compared in Section 4.2. Each experiment has been executed 50 times.

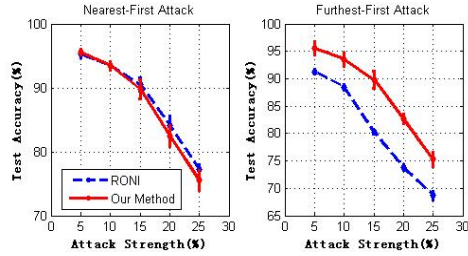
Table 1. Main characteristics of the data sets

Name	# samples	# features
tao	924	2
pim	768	8
h-s	270	13
sonar	208	60

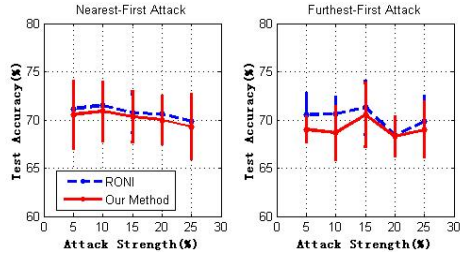
4.1 Accuracy

The average (and also the standard deviation) of the testing accuracies of RONI and the proposed methods under the nearest-First and furthest-first attacks with different attack strengths are shown in Figure 1. The red solid and blue dotted lines represent the result of our method and RONI respectively.

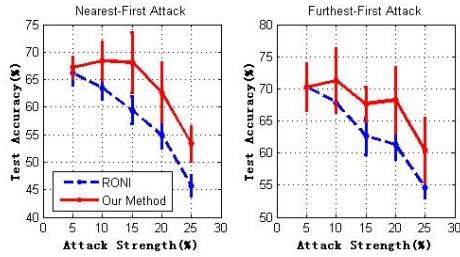
Our method has better performance than RONI in h-s (i.e., 5.99% and 5.30% better in average under nearest-first and furthest-first attack respectively), but in pim, RONI is less than 2% better than our method in all situations. For tao, our method is slightly worse under nearest-first attack when the attack strength increases while under the furthest-first attack, the accuracy of our method is 6.87% higher in average. RONI is worse than our method when the attack strengths are 5% and 10% under the furthest-first attack. When the attack strength values are larger than or equal to 15%, it performs better. On the other hand, our method outperforms in nearest-first attack. There is no difference between the performances of both methods according to the student's t-test with 95% statistical significance. The experimental results suggest that our proposed method achieve the similar performance to the standard RONI.



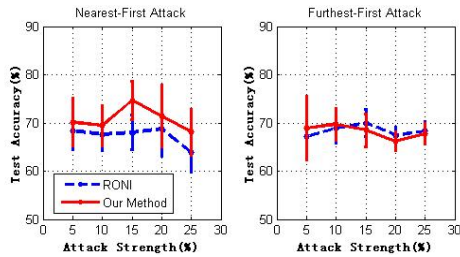
(a) tao



(b) pime



(c) h-s



(d) sonar

Fig. 1. Average and standard deviation of testing accuracy of RONI and the proposed method under the nearest-first attack and the furthest-first attack with different attack strengths (5%, 10%, 15%, 20% and 25%)

4.2 Running Time

The experiments are performed on a PC with 2G of memory and one intel processor with 2 cores at 2.99 GHz. Figure 2 illustrates the average running time of data cleaning using RONI and our proposed method on different datasets.

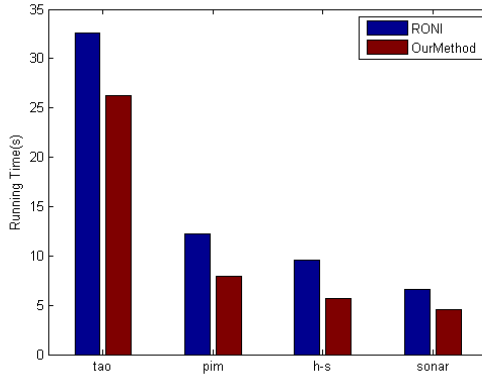


Fig. 2. Running time of RONI and our method (in second)

The results show that the running time of data cleaning of our method is shorter than RONI significantly. As discussed in previous sections, RONI needs to train two classifiers for each sample in the training set. In contrast, the samples are grouped according to the relative neighborhood graph in our method, so the number of iteration is reduced. It explains why our method is more efficient than RONI in terms of time complexity.

5 Conclusions and Future Work

The data sanitization which combines both classifier-independent and classifier-dependent methods is proposed to remove the suspect attack samples. It reduces the time complexity, which is one of the major disadvantages of classifier-dependent method, by considering the structure of the dataset. Different from RONI which evaluates each training sample, our method groups the similar samples as a cluster according to the relative neighborhood graph. All samples in a cluster are evaluated together. As a result, the evaluation number is reduced. Although the experimental results show that the performance of our method and RONI is similar in terms of the testing accuracy of the classifier using the sanitized dataset, the running time of the proposed method significantly smaller than RONI. It suggests that our methods reduce the time complexity of RONI with slightly sacrificing the accuracy.

Acknowledgements. This work is supported by a National Natural Science Foundation of China (61272201), and a Fundamental Research Funds for the Central Universities (10561201465).

References

1. Zhou, Y., Jorgensen, Z., Inge, M.: Combating good word attacks on statistical spam filters with multiple instance learning. Tools with Artificial Intelligence. In: 19th IEEE International Conference on ICTAI 2007, pp. 298–305. IEEE (2007)
2. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure?. In: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, pp. 16–25 (2006)
3. Cretu, G.F., Stavrou, A., Locasto, M.E., Stolfo, S.J.: Casting out demons: Sanitizing training data for anomaly sensors. Security and Privacy. In: IEEE Symposium on SP 2008, pp. 81–95. IEEE (2008)
4. Nelson, B.A.: Behavior of Machine Learning Algorithms in Adversarial Environments. California University Berkeley, Department of Electrical Engineering and Computer Science. No. UCB/EECS-2010-140 (2010)
5. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. Journal of Intelligent Information Systems, pp. 89–109 (2004)
6. Saini, U.: Machine learning in the presence of an adversary: Attacking and defending the spambayes spam filter. California University Berkeley, Department of Electrical Engineering and Computer Science. No. UCB/EECS-2008-62 (2008)
7. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv. 1206.6389 (2012)
8. Lowd, D., Meeck, C.: Good Word Attacks on Statistical Spam Filters. In: CEAS (2005)
9. Sculley, D., Otey, M.E., Pohl, M., Spitznagel, B., Hainsworth, J., Zhou, Y.: Detecting adversarial advertisements in the wild. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 274–282. ACM (2011)
10. Kantchelian, A., Afroz, S., Huang, L., Islam, A.C., Miller, B., Tschantz, M.C., Tygar, J.D.: Approaches to adversarial drift. In: Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security, pp. 99–110. ACM (2013)
11. Xiao, H., Xiao, H., Eckert, C.: Adversarial Label Flips Attack on Support Vector Machines. In: ECAI, pp. 870–875 (2012)
12. Biggio, B., Fumera, G., Roli, F.: Multiple classifier systems for robust classifier design in adversarial environments. International Journal of Machine Learning and Cybernetics 1(1–4), 27–41 (2010)
13. Biggio, B., Corona, I., Fumera, G., Giacinto, G., Roli, F.: Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks. In: Sansone, C., Kittler, J., Roli, F. (eds.) MCS 2011. LNCS, vol. 6713, pp. 350–359. Springer, Heidelberg (2011)
14. Guan, D., Yuan, W., Lee, Y.K., Lee, S.: Nearest neighbor editing aided by unlabeled data. Information Sciences 179(13), 2273–2282 (2009)
15. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. Journal of Intelligent Information Systems 22, 89–109 (2012)
16. Zighed, D.A., Lallich, S., Muhlenbach, F.: Separability Index in Supervised Learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 475–487. Springer, Heidelberg (2002)
17. Toussaint, G.T.: The relative neighborhood graph of a finite planar set. Pattern recognition 12, 261–268 (1980)
18. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.D.: Adversarial machine learning. In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, pp. 43–58. ACM (2011)

19. Biggio, B., Fumera, G., Roli, F.: Design of robust classifiers for adversarial environments. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 977–982. IEEE (2011)
20. Zhou, Y., Jorgensen, Z., Inge, M.: Combating good word attacks on statistical spam filters with multiple instance learning. In: 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, vol. 2, pp. 298–305. IEEE (2007)
21. Rubinstein, B.I., Nelson, B., Huang, L., Joseph, A.D., Lau, S.H., Rao, S., Tygar, J.D.: Antidote: understanding and defending against poisoning of anomaly detectors. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, pp. 1–14. ACM (2009)
22. Barreno, M., Nelson, B., Joseph, A.D., Tygar, J.D.: The security of machine learning. *Machine Learning* **81**(2), 121–148 (2010)
23. Roli, F., Biggio, B., Fumera, G.: Pattern Recognition Systems under Attack. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013, Part I. LNCS, vol. 8258, pp. 1–8. Springer, Heidelberg (2013)
24. UCI: <http://archive.ics.uci.edu/ml/datasets.html>

Author Index

- Adnan, Md. Nasim 219
Ashraful Amin, M. 129
- Beheshti, Maedeh 371
Belaton, Bahari 255
Burns, William 287
- Chan, Patrick P.K. 144, 452
Chang, Huali 24
Chen, Aixia 43
Chen, Guolong 175
Chen, Jiankai 207
Chen, Junfen 255
Chen, Junying 71
Chen, Liming 287
Chen, Qingcai 52, 298, 339
Chooprayoon, Vasin 331
- Deb, Rupam 275
Ding, Hong 442
Donnelly, Mark 287
- Feng, Huimin 43
- Gong, Huachang 161
Gui, Lin 186, 400
Guo, Hongzhi 298
Guo, Zhen 43
- He, Yulin 71
Hou, Yongshuai 52
Hu, Wenxiang 319
Hu, Xiaohui 107
Huang, Qinghua 24
Huang, Yichao 175
- Islam, Md. Zahidul 219
- Jin, Gang 442
- Kumarasinghe, Chithrangi Kaushalya 421
Kwan, Paul W.H. 219
- Li, Anhua 24
Li, Haodi 339
Li, Hongjiang 452
Li, Jie 351
Li, Jun 351
Li, Man 52
Li, Sen 308
Li, Tiechen 107
Li, Yan 431
Li, Ying 431
Li, Yujiao 196
Liao, Xiangwen 175
Liao, Zhengling 107
Liew, Alan Wee-Chung 3, 60, 97, 275
362, 371
Liu, Bin 186, 400
Liu, Lifei 308
Liu, Longzhong 24
Liu, Xiangrui 264
Liu, Ying 351
Lu, Zhihao 107
Lv, Yueming 117
- Ng, Wing W.Y. 117, 161
Nguyen, Mai Phuong 3, 60, 362
Nguyen, Thi Thu Thuy 60
Nguyen, Tien Thanh 3, 60, 97, 362
Nugent, Chris 287
- Pham, Xuan Cuong 362
Poon, Bruce 129
- Qing, Qin 136
Qiu, Qiaoyun 400
Quinn, Susan 287
- Rafferty, Joseph 287
Rahman, Mohammad M. 129
Rong, Tongwen 161
- Shao, Mingwen 264
Shu, Ying 144

- Skillen, Kerry 287
Solheim, Ivar 287
Sun, Chunxiao 298
- Tan, Cong 52
Tang, Bin 186
To, Cuong 97, 362
Tran, Minh Toan 3, 60
Tsang, Eric C.C. 136, 409
- Wang, Donghui 33
Wang, Shanshan 231, 241
Wang, Xiaofen 431
Wang, Xiaolong 52, 339
Wang, Xizhao 33, 207, 231, 241, 388
Wang, Yiding 186
Wei, Jingjing 175
Withanage, D.K. 421
- Xing, Hongjie 308
Xu, Huixin 107
Xu, Ruifeng 186, 400
Xue, Yun 107
- Yan, Hong 129
Yan, Zhengbo 80
Yeung, Daniel S. 196
Yu, Zhiyong 175
- Zeng, Ziqian 117
Zhai, Junhai 33, 207, 231, 241, 319, 379
Zhang, Chengzhi 13
Zhang, Qiangzhi 24
Zhang, Sufang 319, 379
Zhang, Suxian 153
Zhang, Suxiang 153
Zhang, Yao 379
Zhang, Yumin 88
Zhao, Hongya 107, 442
Zhao, Ming 80
Zhao, Shixin 388
Zhao, Suyun 409
Zhou, Liukun 80
Zhou, Qingqing 13
Zhou, Yu 400
Zhu, Hong 33, 231, 241