Hiroshi Motoda Zhaohui Wu Longbing Cao
Osmar Zaiane Min Yao Wei Wang (Eds.)

LNAI 8347

Advanced Data Mining and Applications

9th International Conference, ADMA 2013
Hangzhou, China, December 2013
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Artificial Intelligence 8347

Subseries of Lecture Notes in Computer Science

Hiroshi Motoda Zhaohui Wu Longbing Cao
Osmar Zaiane Min Yao Wei Wang (Eds.)

Advanced Data Mining and Applications

9th International Conference, ADMA 2013
Hangzhou, China, December 14-16, 2013
Proceedings, Part II

Springer

Volume Editors

Hiroshi Motoda
US Air Force Office of Scientific Research, Tokyo, Japan
E-mail: motoda@ar.sanken.osaka-u.ac.jp

Zhaohui Wu
Zhejiang University, Hangzhou, China
E-mail: wzh@cs.zju.edu.cn

Longbing Cao
University of Technology, Sydney, NSW, Australia
E-mail: longbing.cao@uts.edu.au

Osmar Zaiane
University of Alberta, Edmonton, AB, Canada
E-mail: zaiane@cs.ualberta.ca

Min Yao
Zhejiang University, Hangzhou, China
E-mail: myao@zju.edu.cn

Wei Wang
Fudan University, Shanghai, China
E-mail: weiwang1@fudan.edu.cn

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-53916-9 e-ISBN 978-3-642-53917-6
DOI 10.1007/978-3-642-53917-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013956530
CR Subject Classification (1998): I.2, H.3, H.2.8, H.4, I.5, I.4, H.2, J.1
LNCS Sublibrary: SL 7 – Artificial Intelligence

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

Printed on acid-free paper

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

Preface

It is our pleasure to welcome you to the proceedings of the 9th International Conference on Advanced Data Mining and Applications (ADMA 2013).

As the power of generating, transmitting, and collecting huge amounts of data grows continuously, information overload is an imminent problem. It generates new challenges for the data-mining research community to develop sophisticated data-mining algorithms as well as successful data-mining applications. ADMA 2013 was held in Hangzhou, China, with the purpose of promoting original research in advanced data mining and applications and providing a dedicated forum for researchers and participants to share new ideas, original research results, case studies, practical development experiences and applications in all aspects related to data mining and applications, the.

The conference attracted 222 submissions from 26 different countries and areas. All papers were peer reviewed by at least three members of the Program Committee composed of international experts in data-mining fields. The Program Committee, together with our Program Committee Co-chairs, did enormous amount of work to select papers through a rigorous review process and extensive discussion, and finally composed a diverse and exciting program including 32 full papers and 64 short papers for ADMA 2013. The ADMA 2013 program was highlighted by three keynote speeches from outstanding researchers in advanced data mining and application areas: Gary G. Yen (Oklahoma State University, USA), Xindong Wu (University of Vermont, USA), and Joshua Zhexue Huang (Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences).

We would like to thank the support of several groups, without which the organization of the ADMA 2013 would not be successful. These include sponsorship from Zhejiang University, Taizhou University, and University of Technology Sydney. We also appreciate the General Co-chairs for all their precious advice and the Organizing Committee for their dedicated organizing efforts. Finally, we express our deepest gratitude to all the authors and participants who contributed to the success of ADMA 2013.

November 2013

Hiroshi Motoda
Zhaohui Wu
Longbing Cao
Osmar Zaiane
Min Yao
Wei Wang

Organization

ADMA 2013 was organized by Zhejiang University, China

Honorary Chair

Yunhe Pan	Chinese Academy of Engineering, China

Steering Committee Chair

Xue Li	University of Queensland (UQ), Australia

General Co-chairs

Hiroshi Motoda	US Air Force Office of Scientific Research, USA
Zhaohui Wu	Zhejiang University, China
Longbing Cao	University of Technology Sydney, Australia

Program Committee Co-chairs

Osmar Zaiane	University of Alberta, Canada
Min Yao	Zhejiang University, China
Wei Wang	Fudan University, China

Organization Co-chairs

Xiaoming Zhao	Taizhou University, China
Jian Wu	Zhejiang University, China
Guandong Xu	University of Technology Sydney, Australia

Publicity Chair

Liyong Wan	Zhejiang University, China

Registration Chair

Xiaowei Xue	Zhejiang University, China

Web Master

Bin Zeng	Zhejiang University, China

Steering Committee

Kyu-Young Whang	Korea Advanced Institute of Science and Technology, Korea
Chengqi Zhang	University of Technology, Sydney, Australia
Osmar Zaiane	University of Alberta, Canada
Qiang Yang	Hong Kong University of Science and Technology, China
Jie Tang	Tsinghua University, China
Jie Cao	Nanjing University of Finance and Economics, China

Program Committee

Aixin Sun	Nanyang Technological University, Singapore
Annalisa Appice	University Aldo Moro of Bari, Italy
Atsuyoshi Nakamura	Hokkaido University, Japan
Bin Shen	Ningbo Institute of Technology, China
Bo Liu	QCIS University of Technology, Australia
Daisuke Ikeda	Kyushu University, Japan
Daisuke Kawahara	Kyoto University, Japan
Eiji Uchino	Yamaguchi University, Japan
Faizah Shaari	Polytechnic Sultan Salahudin Abddul Aziz Shah, Malaysia
Feiping Nie	University of Texas, Arlington, USA
Gang Li	Deakin University, Australia
Gongde Guo	Fujian Normal University, China
Guandong Xu	University of Technology Sydney, Australia
Guohua Liu	Yanshan University, China
Hanghang Tong	IBM T.J. Watson Research Center, USA
Haofeng Zhou	Fudan University, China
Jason Wang	New Jersey Institute of Technology, USA
Jianwen Su	UC Santa Barbara, USA
Jinjiu Li	University of Technology Sydney, Australia
Liang Chen	Zhejiang University, China
Manish Gupta	University of Illinois at Urbana-Champaign, USA
Mengchu Zhou	New Jersey Institute of Technology, USA
Mengjie Zhang	Victoria University of Wellington, New Zealand
Michael R. Lyu	The Chinese University of Hong Kong, Hong Kong, China
Michael Sheng	The University of Adelaide, Australia
Philippe Fournier-Viger	University of Moncton, Canada
Qi Wang	Xi'an Institute of Optics and Precision Mechanics of CAS, China

Qi Yu	Rochester Institute of Technology, China
Rong Zhu	Jiaxing University, China
Sheng Zhong	SUNY Buffalo, USA
Shu-Ching Chen	Florida International University, USA
Shuliang Wang	Wuhan University, China
Songmao Zhang	Chinese Academy of Sciences, China
Stefano Ferilli	Università di Bari, Italy
Tetsuya Yoshida	Hokkaido University, Japan
Tieyun Qian	Wuhan University, China
Tomonari Masada	Nagasaki University, Japan
Vincent S. Tseng	National Cheng Kung University, Taiwan
Wei Hu	Nanjing University, China
Wei Wang	Fudan University, China
Wynne Hsu	National University of Singapore, Singapore
Xiangjun Dong	Shandong Institute of Light Industry, China
Xiangliang Zhang	King Abdullah University of Science and Technology, Saudi Arabia
Xiao Zheng	Southeast University, China
Xin Jin	University of Illinois at Urbana-Champaign, USA
Ya Zhang	Shanghai Jiao Tong University, China
Yang Gao	Nanjing University, China
Yanglan Gan	Donghua University, China
Yasuhiko Morimoto	Hiroshima University, Japan
Yi Zhuang	Zhejiang GongShang University, China
Yin Song	University of Technology Sydney, Australia
Yong Zheng	DePaul University, USA
Yubao Liu	Sun Yat-Sen University, China
Zhaohong Deng	Alibaba Inc., China
Zhiang Wu	Nanjing University of Finance and Economics, China
Zhihong Deng	Peking University, China
Zhihui Wang	Fudan University, China
Zhipeng Xie	Fudan University, China
Zijiang Yang	York University, Canada

Sponsoring Institutions

College of Computer Science & Technology, Zhejiang University, China
College of Mathematics and information engineering, Taizhou University, China
Advanced Analytics Institute, University of Technology Sydney, Australia

Table of Contents – Part II

Clustering

Association Rule Mining

Pattern Mining

Regression

Prediction

Feature Extraction

Identification

Privacy Preservation

Applications

Machine Learning

Table of Contents – Part I

Opinion Mining

Behavior Mining

Stream Mining

Sequential Data Mining

Web Mining

Image Mining

Text Mining

Social Network Mining

Classification

Semi-supervised Clustering Ensemble Evolved by Genetic Algorithm for Web Video Categorization

Amjad Mahmood, Tianrui Li*, Yan Yang, and Hongjun Wang

School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
amjad.pu@gmail.com, {trli,yyang,wanghongjun}@swjtu.edu.cn

Abstract. Genetic Algorithms (GAs) have been widely used in optimization problems for their high ability in seeking better and acceptable solutions within limited time. Clustering ensemble has emerged as another flavor of optimal solutions for generating more stable and robust partition from existing clusters. GAs have proved a major contribution to find consensus cluster partitions during clustering ensemble. Currently, web video categorization has been an ever challenging research area with the popularity of the social web. In this paper, we propose a framework for web video categorization using their textual features, video relations and web support. There are three contributions in this research work. First, we expand the traditional Vector Space Model (VSM) in a more generic manner as Semantic VSM (S-VSM) by including the semantic similarity between the feature terms. This new model has improved the clustering quality in terms of compactness (high intra-cluster similarity) and clearness (low inter-cluster similarity). Second, we optimize the clustering ensemble process with the help of GA using a novel approach of the fitness function. We define a new measure, Pre-Paired Percentage (PPP), to be used as the fitness function during the genetic cycle for optimization of clustering ensemble process. Third, the most important and crucial step of the GA is to define the genetic operators, crossover and mutation. We express these operators by an intelligent mechanism of clustering ensemble. This approach has produced more logical offspring solutions. Above stated all three contributions have shown remarkable results in their corresponding areas. Experiments on real world social-web data have been performed to validate our new incremental novelties.

Keywords: Genetic Algorithm, Semantic Similarity, Clustering, Clustering Ensemble, Pairwise Constraints, Video Categorization.

1 Introduction

Web video categorization is basically an automatic procedure for assigning web videos to pre-defined categories such as Sports, Autos & Vehicle, Animals,

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 1–12, 2013.

Education, etc. It performs a prominent role in many information retrieval tasks. On social websites (such as YouTube [1]), extreme load of web video data impedes the users to comprehend them properly. Allocation of certain categories to these videos is a primary step. Conventionally, web videos are classified by using audio, textual, visual low-level features or their combinations [2]. These methods depend mostly on building models through machine learning techniques (e.g., SVM, HMM, GMM) to map visual low-level features to the high-level semantics. Due to unsatisfactory results of present high-level concept detection methods [3] and the expense of feature extraction, the content based categorization could not achieve the expected results. In our previous work, we proposed a Semi-supervised Cluster-based Similarity Partitioning Algorithm (SS-CSPA) [4] to categorize the videos containing textual data provided by their up-loaders. We extend this work at two stages, semantic similarity between feature vectors and evolution of the clustering ensemble process with GA.

Semantic similarity plays a significant role in a wide range of data mining applications. Traditional Vector Space Model (VSM) with TF-IDF weighting scheme cannot represent the semantic information of text by neglecting the semantic relevance between the terms. WordNet is an online lexical reference system developed at Princeton University [5]. It attempts to model the lexical knowledge of a native speaker of English. It can also be seen as an ontology for natural language terms. It contains around 100,000 terms, organized into taxonomic hierarchies. As a first addition to our previous work, we developed a new model over WordNet by expanding the VSM in a more generalized way to cater the semantic relation between terms.

GAs are well-known for being highly effective in optimization tasks as well as beneficial in situations where many inputs (variables) interact with one another to produce a large number of possible outputs (solutions). GA formulates the search method that can be used both for solving problems and modeling evolutionary systems. Due to its heuristic nature, no one actually knows in advance, if the solution is totally accurate or not. So, most scientific problems are addressed via estimates, rather than assuming 100% accuracy [6].

So far many contributions have been made to find consensus cluster partitions by GA, however, we propose a new approach based on prior knowledge in terms of must-link information. Using this information, we formulate a new measure, Pre-Paired Percentage (PPP) to define the fitness function during the genetic cycle.

In this paper, we aim to deal with the categorization problem of web videos by using their textual data based on the semi-supervised GA for clustering ensemble. At the same time, in order to improve the quality of base clusters, we extend the traditional VSM using WordNet lexical database support. The rest of the paper is organized as follows. In Section 2, a brief survey of related work is described. Section 3 demonstrates the proposed framework together with the algorithm for web video categorization. Section 4 shows the experimental details along with the evaluation of results. Concluding remarks and future work are stated in Section 5.

2 Related Work

2.1 Web Video Categorization

Automatic categorization of web videos is a crucial task in the field of multimedia indexing. Numerous studies have been conducted so far on this critical subject [2]. Ramchandran et al. [7] proposed a consensus learning approach using YouTube categories for multi-label video categorization. However, the specific categories and the amount of data are not described in their work. Schindler et al. [8] categorized the videos using bag-of-words representation but the classification results are unsatisfactory. Zanetti et al. [9] used 3000 YouTube videos to explore existing video classification techniques. Wu et al. [10] used textual and social information for web video categorization that consists of user upload habits.

2.2 Clustering Ensemble

Recently, semi-supervised clustering ensemble has been proposed and shown a better performance by incorporating the known prior knowledge, e.g., pairwise constraints. Most commonly used constraints are must-link (ML) and cannot-link (CL). A must-link constraint enforces that two objects must belong to the same cluster while a cannot-link constraint enforces that two objects must belong to different clusters [11]. Zhou et al. [12] proposed a disagreement-based semi-supervised learning paradigm, where multiple learners are trained for the task and the disagreements among the learners are exploited during the semi-supervised learning process. Zhou et al. [13] pointed out that most semi-supervised ensemble methods work by training learners using the initial labeled data first, and then using the learners to assign pseudo-labels to unlabeled data. Wang et al. [14] explored a semi-supervised cluster ensemble model based on semi-supervised learning and ensemble learning utilizing Bayesian network and EM algorithm. Yang et al. [15] proposed a new constrained Self-Organizing Map (SOM) to combine multiple semi-supervised clustering solutions for further enhancing the performance of ICop-Kmeans in intelligent decision support systems. Yang et al. [16] presented a novel semi-supervised consensus clustering ensemble algorithm based on multi-ant colonies.

2.3 Semantic Similarity

Semantic similarity is related to computing the similarity between concepts which are not lexicographically similar. Most popular semantic similarity methods are implemented and evaluated using WordNet as the underlying reference ontology. The initial research efforts by Leacock et al. [17] and Wu et al. [18] in this area are based on path lengths between pair of concepts. Leacock et al. find the shortest path between two concepts, and scales that value by the maximum path length found in the is–a hierarchy in which they occur. Wu et al. find the depth of the least common subsumer (LCS) of the concepts, and then

scales that by the sum of the depths of the individual concepts. The depth of a concept is simply its distance to the root node. The measure path is a baseline that is equal to the inverse of the shortest path between two concepts. WordNet similarity has been used by a number of other researchers in an interesting array of domains. Zhang et al. [19] used it as a source of semantic features for identifying crossdocument structural relationships between pairs of sentences found in related documents. McCarty et al. [20] used it in conjunction with a thesaurus derived from raw text in order to automatically identify the predominant sense of a word. Baldwin et al. [21] used WordNet similarity to provide an evaluation tool for multiword expressions that are identified via Latent Semantic Analysis.

2.4 Genetic Algorithm

GA has been widely and successfully used in a number of research areas, but here we have most concerns with its application in clustering ensemble. Azimi et al. [22] used intelligent mutation with one point crossover to achieve fast convergence, simplicity, robustness and high accuracy. Yoon et al. [23] generated different types of multi source data through a variety of different experiments and applied GA to generate better cluster results than those obtained using just one data source. Ramanathan and Guan [24] involved a hybrid combination of a global clustering algorithm followed by a corresponding local clustering algorithm. Faceli et al. [25] considered the knowledge of some exiting complete classification of such data based on Multi Objective Clustering Ensemble Algorithm (MOCLE). Hong and Kwong [6] used Genetic-guided Clustering algorithm with Ensemble Learning operator (GCEL) to achieve a comparative or better clustering solution with less fitness evaluations. Ozyer and Alhajj [26] solved the scalability problem by applying the divide-and-conquers approach in an iterative way to handle the clustering process.

3 Proposed Framework

3.1 System Overview

As stated earlier, this work is an extension of our previous research [4], where we proposed Semi-supervised Cluster-based Similarity Partitioning Algorithm (SS-CSPA), so the main framework is the same as previous. The two new additions are *Semantic Vector Space Model (S-VSM)* and the evolution of clustering ensemble process with the help of GA. Textual features of YouTube videos are used for feature term vector representation after basic pre-processing. S-VSM generates the similarity matrix to be used for clustering purpose. Here we select three algorithms, graph partitioning, spectral clustering and affinity propagation [27] for the clustering purpose. Related videos information is translated into must-link constraints. Clusters are ensembled with CSPA, MCLA and HGPA [28] in a genetic cycle with a fitness function based on our new measure PPP. The overall scheme is bundled into a Semi-supervised Cluster-based Similarity Partitioning Algorithm evolved by GA (SS-CSPA-GA). The framework of the proposed SS-CSPA-GA algorithm is shown in Fig. 1.

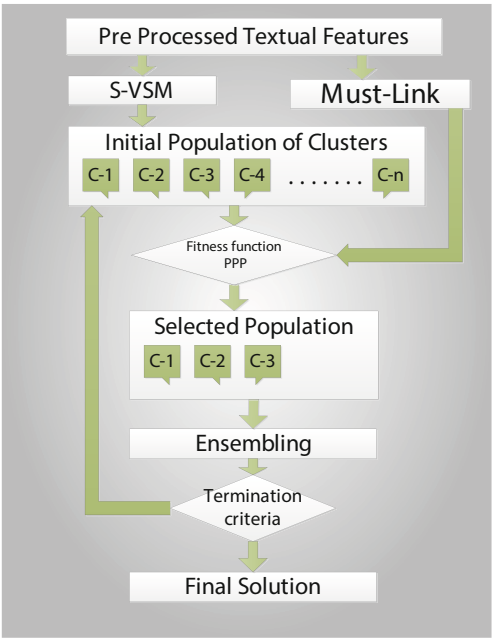


Fig. 1. The proposed framework

3.2 Semantic Vector Space Model (S-VSM)

In traditional vector space model, a simple two-fold heuristics [29] based on frequency is used to score each component directly as a function of *Term Frequency* (TF) referring to the number of occurrences of a particular term in a specific document, and *Inverse Document Frequency* (IDF) refering to the distribution of a particular term across all documents. The basic theme of $TF-IDF$ scheme is that, if a word appears frequently in a document, it must be an important keyword, unless it also appears frequently in other documents. The similarity measure for two documents D_i and D_j, can be calculated by using normalized Cosine function between them,

$$D_i = (w_{i1}, w_{i2}, ...w_{iN}) \text{ and } D_j = (w_{j1}, w_{j2}, ...w_{jN}). \tag{1}$$

$$Sim(D_i, D_j)_{TS} = \frac{\sum_{t=1}^{N}(w_{it} * w_{jt})}{\sqrt{\sum_{i=1}^{N}(w_{it})^2 * \sum_{t=1}^{N}(w_{jt})^2}}. \tag{2}$$

The most important factor at this stage is VSM does NOT consider any relationship between feature terms. All terms are considered as independent of each other. In our new model, we find the semantic relationship between the feature terms with the help of WordNet and propose the new similarity measure with the mesh topology framework.

$$Sim(D_i, D_j)_{SS} = \frac{\sum_{x=1}^{f_n}[\sum_{y=1,y\neq x}^{f_n}(w_{ix} * w_{jy} + w_{jx} * w_{iy})F_{xy}]}{\sum_{x=1}^{f_n}[\sum_{y=1,y\neq x}^{f_n}(w_{ix} * w_{jy} + w_{jx} * w_{iy})]}. \quad (3)$$

where F_{xy} = Terms relevance matrix obtained from WordNet and f_n = Total no of terms. The final similarity between two documents will be the addition of Eq. 2 and Eq. 3.

$$Sim(D_i, D_j) = Sim(D_i, D_j)_{TS} + Sim(D_i, D_j)_{SS} \quad (4)$$

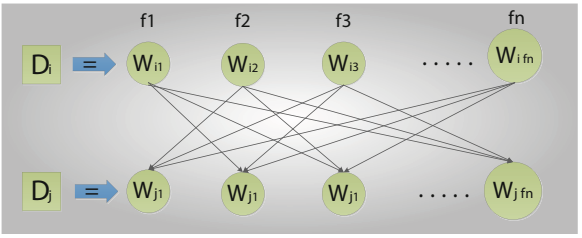


Fig. 2. Semantic Vector Space Model

3.3 Genetic Algorithm

In this section, we discuss the main features of GA, including chromosome representation, population initialization, fitness function, crossover and mutation operators.

Genotype: In our framework, we propose a string type chromosome encoding consisting of 4 groups of information in 8 bits. So our initial solution space comprises of 256 candidates, as shown in Table 1.

Table 1. Genotype Encoding

Choromosome	1	2	3	4	5	6	7	8	9
Information	GA	Data Subset		Must Link		Clustering		Ensembling	

The first bit represents that if a solution is originated from the base assembling process or through a genetic cycle. Next two bits represent the dataset (Title, Tag, Description or All). Next two bits represent which configuration of must-link (ML-1, ML-2, ML-3 or ML-4) is used during semi-supervised clustering ensemble process. Last two bits represent which clustering ensemble algorithm (CSPA, MCLA or HGPA) is used.

Fitness Function: We propose to translate the related video information in terms of must-link constraints. We define a new measure PPP. It is the percentage of must-link pairs in a clustering result that satisfy must-link rule, i.e., both members of the pair exists in the same cluster. Now during the selection of suitable candidates for ensemble process, the clustering candidates are evaluated according to this measure. The idea is that a group of cluster with a high value of PPP measure is expected to produce more accurate results as compared to another group with a low value of PPP measure.

Termination Criteria: The validated datasets are used to estimate the accuracy of clustering ensemble which ultimately reflects the performance of genetic cycle. The resulting labels are compared with ground truth labels using Micro-Precision (MP) [31] accuracy measure. To terminate the genetic cycle, number of iterations (N_{GC}) and saturated (A difference between two consecutive cycles is negligible) MP accuracy (S_{MP}) are monitored on *Whichever Comes First* policy basis.

Genetic Operator: Crossover and mutation are the most important steps in GA. The implementation of these operators is a critical process. We propose to represent these operators by an intelligent mechanism of clustering ensemble.

3.4 The Algorithm

The proposed SS-CSPA-GA Algorithm is outlined as follows.

Input: (i) Dataset, containing textual part of videos (UtVd).
(ii) Related Video information (RVi).
(iii) Validated Video Dataset (VVD).
Output: Clustering labels.

Calculate all possible configurations of pairwise constraints M-(1,2,3,4) from RVi.
Concatenate the validated dataset (VVD) with testing dataset(UtVd).
for $i \in \{DataSets\ UtVd\ (A1, A2, A3)\}$
  for $j \in \{DataSet\ Copies\ C_j\}$
    for $k \in \{Title, Tag, Description\}$
      Text pre-processing for extraction of unique and meaningful terms.
      Apply $TF - IDF$ scheme to find term weights.
      Calculate the initial similarity matrix S_{TS} and list of feature terms.
      Calculate the semantic relevance of feature terms.
      Calculate the semantic similarity matrix S_{SS}.
      Add two similarity matrices to get the final similarity S.
        for $m \in \{M - 0, M - (1, 2, 3, 4)\}$ M-0 is without must-link

Calculate net similarity $S_n = S_i + m$.
Execute different clustering algorithms and get labels.
Calculate PPP measure for every cluster label.
End m
End k
End j
End i
While (N_{GC} *or* S_{MP})
Select a group of clusters with high (**Top x**) PPP measure.
Ensemble the clustering Labels to get a new solution.
End While

---

4 Experiments

4.1 Datasets and Evaluation Criteria

Datasets Among the different available datasets, we select MCG-WEBV [30] benchmark dataset including 80,0311 most viewed videos for every month from Dec. 2008 to Feb. 2009 on YouTube. These videos are most valuable to do mining for their high quality and popular contents by providing comprehensive features for the video analysis and process. It includes the raw videos, keyframes, five metadata features, eight web features, and eleven low-level features cover textual, visual and audio.

We perform a number of experiments on textual part by considering the basic textual features like title, tag and description. Related videos data is also included as a must-link constraint. Some basic facts about three considerable datasets are stated in Table 2.

Table 2. Dataset description

DataSet UtVd	**Number of Categories**	**Copies** C_i	**Instances**	**Features** Title	Tag	Des
A-1	8,9,14	2	1007	2102	4250	5420
A-2	1,12,13	2	1010	1697	3569	4927
A-3	2,6,7	2	995	1783	3867	5197

Validated Dataset. For each testing dataset UtVd, we select about 20% in size, a set of additional pre-labeled videos. During the genetic cycle, the termination criteria is determined by comparing the resulting labels of validated datset with their corresponding gound truth labels.

Evaluation Criteria. For the final evaluation of results, we use micro-precision [31] to measure the accuracy of the consensus cluster with respect to the true labels. The micro-precision is defined as

$$MP = \sum_{h=1}^{K} [\frac{a_h}{n}], \tag{5}$$

where K is the number of clusters and n is the number of objects, a_h denotes the number of objects in consensus cluster h that are correctly assigned to the corresponding class. We identify the corresponding class for consensus cluster h as the true class with the largest overlap with the cluster, and assign all objects in cluster h to that class. Note that $0 \leq MP \leq 1$ with 1 indicating the best possible consensus clustering which has to be in full agreement with the class labels.

4.2 Results

Using the above stated scheme, we first perform the three clustering algorithms with must-link constrains and find the clustering labels. For each dataset, we select at least three subsets for experiments and take their average. The results are compared with true labels to find the accuracy. The average accuracy for three datasets with spectral clustering is shown in Table 3.

Table 3. Average clustering accuracies for three datasets with spectral clustering

Data	**M-0**	**M-1**	**M-2**	**M-3**	**M-4**	**M-0**	**M-1**	**M-2**	**M-3**	**M-4**	**M-0**	**M-1**	**M-2**	**M-3**	**M-4**
SubSection	**Dataset A-1**					**Dataset A-2**					**Dataset A-3**				
Title	0.82	0.89	0.92	0.76	0.78	0.53	0.53	0.72	0.62	0.59	0.59	0.50	0.65	0.59	0.73
Tag	0.92	0.94	0.95	0.94	0.93	0.83	0.83	0.85	0.71	0.84	0.67	0.73	0.77	0.75	0.71
Des	0.64	0.92	0.71	0.75	0.77	0.63	0.59	0.62	0.60	0.70	0.47	0.41	0.73	0.42	0.66
All	0.82	0.95	0.95	0.89	0.93	0.80	0.81	0.85	0.65	0.78	0.75	0.76	0.71	0.72	0.72

Considering the above results as base clusters, we execute three clustering ensemble algorithms, CSPA, MCLA and HGPA. The best results are obtained from CSPA as shown in Table 4.

Table 4 shows clearly that the clustering ensemble with genetic guidance has evolved towards better solution. In each data set $A - i$, the results in bold face values (GA bit = 1) are obtained through the genetic cycle, clearly represents the evolution of better solution during genetic cycle with the guidance of musk-link constraints formulating the PPP measure. At the same time, the fitness function measure makes it easy and efficient to select best individuals from the population solution.

Table 4. Clustering ensemble for datasets in the genetic cycle

Dataset A-1		**Dataset A-2**		**Dataset A-3**	
Choromosome	Accuracy	Choromosome	Accuracy	Choromosome	Accuracy
0 01 10 00 00	0.94	0 01 01 00 00	0.87	0 11 00 00 00	0.83
0 10 00 00 00	0.95	0 01 11 00 00	0.88	0 01 10 00 00	0.84
0 11 11 00 00	0.96	0 11 00 00 00	0.89	0 01 01 00 00	0.85
1 00 00 00 00	**0.97**	1 00 00 00 00	**0.90**	1 00 00 00 00	**0.86**
1 00 00 00 00	**0.98**	1 00 00 00 00	**0.91**	1 00 00 00 00	**0.87**
1 00 00 00 00	**0.99**	1 00 00 00 00	**0.92**	1 00 00 00 00	**0.88**

4.3 Results Discussion

1. As stated earlier, this work is an extension of our previous work in two dimensions.
2. First idea is the generalization of traditional VSM. As this concept is more natural and logical, it produces better results in terms of the clustering quality. Our source in this generalization is WordNet database. Another thought can be to use some other external source like ImageNet database.
3. The second part of this work is the evolution of clustering ensemble process with the help of GA. Although researches has already proposed many algorithms in this context, but still our approach is novel for the configuration of GA like, the use of must-link information in the fitness function and the representation of genetic operators by the clustering ensemble.
4. The available noisy text information and less dense constrains are not sufficient to fully categorize the videos data. There is a need of some more data sources like user interest videos, visual contents of corresponding videos and specifically some more external information retrieval sources like ImageNet database. Merging of such external information support is also a challenge.

5 Conclusions

This paper proposed a novel approach, SS-CSPA-GA, to categorize the videos containing textual data provided by their up-loaders. Experimental results showed that the proposed two extensions worked well for categorization purpose. The categories of our dataset (YouTube) are not very well distinguished. There is an overlap in many categories like Music with Film & animation, Education with Science & Technology and Comedy with Entertainment. So instead of considering them as independent categories, a hierarchical tree relationship of these categories can be a more natural and logical design (future work consideration). This idea can place new challenges for data mining and multimedia researches. We use must-link information in our fitness function. Another idea can be to use cannot-link information if available. In our future work, while searching for more supportive information, the fusion of additional information for web video categorization will also be a challenge.

Acknowledgement. This work was supported by the National Science Foundation of China (Nos. 61175047, 61170111, 61262058 and 61003142), the Fundamental Research Funds for the Central Universities (Nos. SWJTU11ZT08 and SWJTU12CX092) and the Research Fund of Traction Power State Key Laboratory, Southwest Jiaotong University (No. 2012TPL_T15).

References

1. YouTube, `http://www.youtube.com`
2. Brezeale, D., Cook, D.J.: Automatic video classification: A survey of the literature. IEEE Transactions on Systems, Man, and Cybernetics, 416–430 (2008)
3. Jiang, Y.-G., Ngo, C.-W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proceedings of 6th ACM International Conference on Image and Video Retrieval, pp. 494–501. ACM, New York (2007)
4. Mahmood, A., Li, T., Yang, Y., Wang, H., Afzal, M.: Semi-supervised Clustering Ensemble for Web Video Categorization. In: Zhou, Z.-H., Roli, F., Kittler, J. (eds.) MCS 2013. LNCS, vol. 7872, pp. 190–200. Springer, Heidelberg (2013)
5. WordNet by Princeton, `http://wordnet.princeton.edu`
6. Hong, Y., Kwong, S.: To combine steady-state genetic algorithm and ensemble learning for data clustering. Pattern Recogn. Lett. J. 29(9), 1416–1423 (2008)
7. Ramachandran, C., Malik, R., Jin, X., Gao, J., Nahrstedt, K., Han, J.: Videomule: A consensus learning approach to multi-label classification from noisy user-generated videos. In: Proceedings of 17th ACM International Conference on Multimedia, pp. 721–724. ACM, New York (2009)
8. Schindler, G., Zitnick, L., Brown, M.: Internet video category recognition. In: Proceedings of Computer Vision and Pattern Recognition Workshops, Atlanta. Georgia Institute of Technology, pp. 1–7 (2008)
9. Zanetti, S., Zelnik-Manor, L., Perona, P.: A walk through the web's video clips. In: Proceedings of Computer Vision and Pattern Recognition Workshops, Pasadena. California Institute of Technology, pp. 1–8 (2008)
10. Wu, X., Zhao, W.L., Ngo, C.-W.: Towards google challenge: combining contextual and social information for web video categorization. In: Proceedings of 17th ACM International Conference on Multimedia, pp. 1109–1110. ACM, New York (2009)
11. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: International Conference on Machine Learning, New York, pp. 577–584 (2001)
12. Zhou, Z.-H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems, 415–439 (2010)
13. Zhou, Z.-H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton (2012)
14. Wang, H., et al.: Semi-Supervised Cluster Ensemble Model Based on Bayesian Network. Journal of Software 21(11), 2814–2825 (2010) (in Chinese)
15. Yang, Y., Tan, W., Li, T., Ruan, D.: Consensus Clustering Based on Constrained Self-Organizing Map and Improved Cop-Kmeans Ensemble in Intelligent Decision Support Systems. Knowledge-Based Systems 32, 101–115 (2012)
16. Yang, Y., Wang, H., Lin, C., Zhang, J.: Semi-supervised clustering ensemble based on multi-ant colonies algorithm. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS (LNAI), vol. 7414, pp. 302–309. Springer, Heidelberg (2012)

17. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database 49, 265–283 (1998)
18. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133–138 (1994)
19. Zhang, Z., Otterbacher, J., Radev, D.: Learning cross-document structural relationships using boosting. In: Proceedings of the 12th International Conference on Information and Knowledge Management, pp. 124–130 (2003)
20. McCarthy, D., Koeling, R., Weeds, J.: Ranking WordNet senses automatically. Technical Report CSRP 569, University of Sussex (2004)
21. Baldwin, T., Bannard, C., Tanaka, T., Widdows, D.: An empirical model of multiword expression decomposability. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 89–96 (2003)
22. Azimi, J., Mohammadi, M., Movaghar, A., Analoui, M.: Clustering ensembles using genetic algorithm. In: Proceedings of the International Workshop on Computer Architecture for Machine Perception and Sensing, pp. 119–123. IEEE (2007)
23. Yoon, H.-S., Lee, S.-H., Cho, S.-B., Kim, J.H.: Integration analysis of diverse genomic data using multi-clustering results. In: Maglaveras, N., Chouvarda, I., Koutkias, V., Brause, R. (eds.) ISBMDA 2006. LNCS (LNBI), vol. 4345, pp. 37–48. Springer, Heidelberg (2006)
24. Ramanathan, K., Guan, S.-U.: Recursive self-organizing maps with hybrid clustering. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, pp. 1–6 (2006)
25. Faceli, K., de Carvalho, A.C.P.L.F., de Souto, M.C.P.: Multi-objective clustering ensemble with prior knowledge. In: Sagot, M.-F., Walter, M.E.M.T. (eds.) BSB 2007. LNCS (LNBI), vol. 4643, pp. 34–45. Springer, Heidelberg (2007)
26. Ozyer, T., Alhajj, R.: Parallel clustering of high dimensional data by integrating multi-objective genetic algorithm with divide and conquer. Applied Intelligence 31, 318–331 (2009)
27. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315(5814), 972–976 (2007)
28. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
29. Salton, G., Buckley, C.: Term-weighing approache sin automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)
30. Cao, J., Zhang, Y.-D., Song, Y.-C., Chen, Z.-N., Zhang, X., Li, J.-T.: MCG-WEBV: A Benchmark Dataset for Web Video Analysis. Technical Report, ICT-MCG-09-001 (2009)
31. Zhou, Z., Tang, W.: Clusterer ensemble. Knowledge Based System 19(1), 77–83 (2006)

A Scalable Approach for General Correlation Clustering

Yubo Wang, Linli Xu, Yucheng Chen, and Hao Wang

School of Computer Science and Technology,
University of Science and Technology of China, Anhui, China
linlixu@ustc.edu.cn,
{wybang,ycchen,xdwangh}@mail.ustc.edu.cn

Abstract. We focus on the problem of correlation clustering, which is to partition data points into clusters so that the repulsion within one cluster and the attraction between clusters could be as small as possible without predefining the number of clusters k. Finding the optimal solution to the problem is proven to be NP-hard, and various algorithms have been proposed to solve the problem approximately. Unfortunately, most of them are incapable of handling large-scale data. In this paper, we relax the problem by decoupling the affinity matrix and cluster indicator matrix, and propose a pseudo-EM optimization method to improve the scalability. Experimental results on synthetic data and real world problems including image segmentation and community detection show that our technique achieves state of the art performance in terms of both accuracy and scalability.

Keywords: Correlation clustering, Unsupervised learning, Large scale, Pseudo-EM algorithm.

1 Introduction

Clustering is one of the most fundamental problems in machine learning, with the goal to partition data points into groups such that the points within clusters are more similar to each other than those in different clusters. The clustering problem has received a significant amount of attention during the past few decades, and numerous methods have been proposed to solve it. However, most of them need the number of clusters k as a priori. *Correlation Clustering* [1] makes an exception, which is able to select k automatically. Moreover, this "model selection" property can be theoretically justified with a probabilistic interpretation [2], and theoretical analysis has been conducted for correlation clustering with error bounds derived [3].

Correlation clustering is a graph-based problem, where vertices correspond to the data points, and each edge (u,v) is labeled either "+" or "-" depending on whether vertices u and v are similar or not. Given this complete binary affinity graph, the task of correlation clustering is to minimize the "-" edges (repulsion) within clusters and "+" edges (attraction) between clusters, which is also known as minimizing disagreements. An equivalent optimization problem is to maximize agreements — maximize the "+" edges within clusters and "-" edges between clusters. The correlation clustering problem is proven to be NP-complete [1], and the majority of efforts are then devoted to solving it approximately [1, 4–7]. Among them, convex continuous relaxation is

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 13–24, 2013.

frequently applied. A linear programming (LP) formulation in [4] results in a factor 4-approximation algorithm for minimizing disagreements. For maximizing agreements, several relaxations based on semi-definite programming (SDP) are achieved [4, 6]. To make the problem more flexible, [4, 5] extend the binary graphs to general weighted graphs, which contain both positive and negative edges.

Despite the large amount of theoretical analysis conducted on correlation clustering, most of the existing algorithms are impractical for real-world applications which are relatively large-scale [8, 9]. For example, there are $O(n^3)$ constraints in the LP relaxation for minimizing disagreements, while the SDP formulation with $O(n^2)$ variables for maximizing agreements is known to be computationally expensive and hard to scale up. Some recent work has focused on the computational issue and tried to address it. In [10], a more effective relaxation is proposed by exploiting the special problem domain. Discrete energy minimization algorithms are adopted in [2] to scale up the computational procedure. Although these approaches do improve the efficiency of correlation clustering, they are still insufficient for real-world problems.

In this paper, we reformulate correlation clustering with a new perspective of decoupling the affinity matrix and cluster indicator matrix. A pseudo-EM optimization method is proposed by relaxing the new formulation. Beyond that, to further improve the performance, we adopt online updates and extend the algorithm to adapt to sparse data by appending a sparsity factor. Experiments are performed on synthetic data and practical tasks including pixel-level image segmentation and community detection in real information networks, and convincing results are achieved to demonstrate the advantages of our proposed technique in terms of both accuracy and scalability.

The remainder of the paper is organized as follows. In the next section we give a brief introduction to the correlation clustering problem. In Section 3 we show how to reformulate and solve it with effective alternating minimization routine. Experimental results are presented to demonstrate the effectiveness of the proposed algorithms in Section 4 and in Section 5 we conclude the paper with possible directions for future work.

2 Correlation Clustering

Correlation clustering is defined on a complete graph $G = (V, E)$, with n vertices corresponding to the data points to be clustered and an edge between every pair of nodes. Each edge is assigned with a label $e(u, v) \in \{+, -\}$ where $e(u, v) = +$ if u and v are similar, $e(u, v) = -$ otherwise. In this paper, we will focus on the minimizing disagreements objective of correlation clustering.

Assuming cluster assignments can be represented with natural numbers, the goal of correlation clustering is to find a cluster assignment $\mathcal{C} : V \to \mathbb{N}$ by solving the following problem: $\min_{\mathcal{C}} \sum_{e(u,v)=+} 1[\mathcal{C}(u) \neq \mathcal{C}(v)] + \sum_{e(u,v)=-} 1[\mathcal{C}(u) = \mathcal{C}(v)]$. One can further extend the complete graph with binary affinity to a general graph. This graph can be described with an affinity matrix $W \in \mathbb{R}^{n \times n}$:

$$W \begin{cases} > 0 : u \text{ and } v \text{ attract each other by } |W_{uv}| \\ < 0 : u \text{ and } v \text{ repel each other by } |W_{uv}| \\ = 0 : \text{the relation between } u \text{ and } v \text{ is uncertain} \end{cases},$$

and the clustering objective for the general graph can be written as

$$\min_{\mathcal{C}} \sum_{W_{uv}>0} 1[\mathcal{C}(u) \neq \mathcal{C}(v)]W_{uv} - \sum_{W_{uv}<0} 1[\mathcal{C}(u) = \mathcal{C}(v)]W_{uv} \ .$$

By introducing a matrix D in which $D_{uv} = 1$ if u and v are in the same cluster and $D_{uv} = -1$ otherwise, we can notice that D encodes an equivalence relation, namely that it is transitive, reflexive and symmetric. Thus the minimizing disagreements problem can be rewritten as

$$\begin{aligned} \min_{D} \quad & -\Big(\sum_{\substack{W_{uv}>0 \\ D_{uv}<0}} W_{uv}D_{uv} + \sum_{\substack{W_{uv}<0 \\ D_{uv}>0}} W_{uv}D_{uv}\Big) \\ \text{s.t.} \quad & D_{uv} \in \{-1, 1\}, \forall u, v; \ D_{uu} = 1, \forall u; \\ & D_{uv} = D_{vu}, \forall u, v; \ D_{uv} + D_{vs} \leq D_{us} + 1, \forall u, v, s \end{aligned} \quad . \tag{1}$$

One should notice that any feasible D matrix in (1) corresponds to an equivalence relation, and therefore it is straightforward to recover a clustering from the solution to (1).

Correlation Clustering Optimization. Solving (1) exactly is NP-complete [1]. A natural way out is then to relax the hard equivalence relation constraints on D. Actually, simply by relaxing the discrete constraints $D \in \{-1, 1\}^{n\times n}$ to be continuous, the problem can be reformulated as a linear program with $O(n^3)$ linear constraints [4, 5]. Due to the cubic number of constraints, the time complexity of the LP formulation grows rapidly with the problem size.

Another relevant piece of work is to start with maximizing agreements problem with similar objective and constraints, and apply semi-definite relaxation to a linear transformation of the D variable. As a consequence, a semi-definite optimization problem with an $n \times n$ matrix variable and $O(n^2)$ linear constraints can be formulated. As a convex optimization problem, it can be solved with polynomial time. However the time complexity of solving an SDP with a matrix variable of size p and q constraints is up to $O(q^2p^{2.5})$ [11], which is prohibitive for even medium-size data.

From the discussion above, we can conclude that although convex relaxations have the nice property that global optimum can be found for the relaxed problems in polynomial time, unfortunately they are not practical for large-scale problems.

Another possible direction is to trade convexity for scalability. Recently, [2] takes a new perspective and treats correlation clustering optimization as a special discrete energy minimization problem without unary terms. Based on techniques in energy minimization [12, 13], several algorithms are proposed including *Expand-and-Explore*, *Swap-and-Explore* and *Adaptive-label ICM*. Compared to the continuous convex relaxations discussed above, significant improvements in scalability are achieved in this framework, which are chosen as the rival algorithms in our experiments.

3 Pseudo-EM Algorithm

Instead of the indirect routine of solving correlation clustering by first computing the relaxed cluster equivalence relation matrix D and then recovering the cluster assignments

based on D, we take a more intuitive and straightforward perspective, which is to fixate on the cluster label assignments directly.

We first define a *clustering indicator matrix* L which describes the cluster assignments of the vertices. Specifically, $L_{iu} = 1$ means that vertex u is in cluster i, $L_{iu} = -1$ otherwise. L encodes a valid clustering if it satisfies the following: a data point belongs to one and only one cluster; each cluster contains at least one data point. That is,

$$\mathcal{Q} : \{L \in \{-1, 1\}^{k \times n};\ \sum_{i=1}^{k} L_{iu} = 2 - k, \forall u;\ \sum_{u=1}^{n} L_{iu} > -n, \forall i\}\ , \tag{2}$$

where k is the number of clusters and not predefined in correlation clustering.

Proposition 1. *The correlation indicator vector $D_{u,:}$ is the same as the cluster indicator vector $L_{\mathcal{C}(u),:}$ for any vertex u, where $\mathcal{C}(u)$ denotes the cluster assignment of u.*

Based on Proposition 1, it is possible to replace D with L in (1) and solve for L directly. Moreover, W and L can be treated as general variables without any inherent physical meaning in the new objective. As a result, a new reformulation and corresponding relaxation by decoupling W and L are derived with a two-step effective alternating optimization method, which is similar to expectation-maximization (EM) algorithm: compute a latent variable $\mathcal{C}$ based on L in the first step and optimize L according to $\mathcal{C}$ in the following step.

3.1 The Basic Pseudo-EM Routine

Consider the correlation clustering problem in a general graph (1). According to Proposition 1, the following relations are true:

$$\begin{aligned}
&D_{uv}, \forall u, v \quad &\Longleftrightarrow \quad &L_{\mathcal{C}(u)v}, \forall \mathcal{C}(u) \in \{1, 2, ..., k\}, \forall v;\\
&D_{uu} = 1, \forall u \quad &\Longleftrightarrow \quad &L_{\mathcal{C}(u)u} = 1, \forall u;\\
&\{D_{uu} = 1, \forall u;\ D_{uv} = D_{vu}, \forall u, v;\ D_{uv} + D_{vs} \leq D_{us} + 1, \forall u, v, s\} \Longleftrightarrow \quad &\mathcal{Q}\ .
\end{aligned}$$

Thus we can replace D with L according to the equivalence mentioned above and rewrite (1) as a summation of loss produced by each row of W and L, which can be expressed as

$$\begin{aligned}
\min_{L, \mathcal{C}, k} \quad & -\sum_{u} \sum_{v \atop W_{uv} L_{\mathcal{C}(u)v} < 0} W_{uv} L_{\mathcal{C}(u)v} \\
\text{s.t.} \quad & \mathcal{C}(u) \in \{1, 2, ..., k\}, \forall u;\ \mathcal{Q};\ L_{\mathcal{C}(u)u} = 1, \forall u
\end{aligned}\ . \tag{3}$$

Notice in the optimization problem above, apart from the variables including the cluster indicator matrix L and the number of clusters k, we introduce an auxiliary variable $\mathcal{C}$ which corresponds to the cluster assignment. These variables are coupled with each other through the constraints. Despite the redundancy of the variables, we benefit from treating them separately in the optimization procedure as we will see shortly.

The reformulated problem is still computationally hard to solve. However, we can first relax the constraint $L_{\mathcal{C}(u)u} = 1$ (which means u is in cluster $\mathcal{C}(u)$ physically) and fix L, leaving $\mathcal{C}$ the only variable to be optimized. In this way, we can minimize the objective in an iterative manner: assume at iteration t, we have a feasible candidate for L: $L^t = [\ell_1, \ell_2, ..., \ell_k]$. By fixing L^t, we can optimize over $\mathcal{C}$, where the best $\mathcal{C}^{t+1}(u)$ for vertex u can be simply computed by enumerating all possible class assignments for u, and solve the following optimization problem:

$$\mathcal{C}^{t+1}(u)^* = \underset{\mathcal{C}(u)\in\{1,2,...,k\}}{\arg\min} \sum_{v, W_{uv} L^t_{\mathcal{C}(u)v} < 0} -W_{uv} L^t_{\mathcal{C}(u)v} \ . \tag{4}$$

Once the cluster assignment for all the vertices $\mathcal{C}$ is completed, it is straightforward to update L accordingly: we first set k to the number of unique clusters in $\mathcal{C}$ and reassign $\mathcal{C}$ to take values from $\{1, ..., k\}$. L can then be updated by starting from a $k \times n$ matrix of all -1's and setting $L_{\mathcal{C}(u)u} = 1$. This optimization routine works like expectation-maximization (EM) algorithm in the sense that it computes the latent variable $\mathcal{C}$ in the first step and optimizes L in the second step, therefore we call it pseudo-EM. Notice that it is possible some clusters contain no vertices at some iteration, and they would disappear after the iteration, which makes the number of clusters be selected to adapt to a lower loss. To improve the convergence rate and optimization quality, this procedure can be conducted in an online way, that means when deciding the cluster assignments for new vertices, the existing assignments are already in effect. This iterative procedure is shown in Algorithm 1.

Given a candidate for L, Algorithm 1 chooses the best cluster assignment for every u in each iteration to minimize the objective, and therefore the loss will decrease until convergence. Due to the property of selecting k automatically in Algorithm 1, there is no need to predefine the number of clusters. As we can observe from (4), vertices with similar affinity vectors will be likely in the same cluster. This implies that although we remove the constraint $L_{\mathcal{C}(u)u} = 1$, which is equivalent to the strong equivalence relation encoded in D, the nature of correlation clustering is preserved due to the intuitive optimization procedure.

Theorem 1. *Algorithm 1 is guaranteed to converge.*

Proof. First notice that in each iteration $L^t = [\ell_1, \ell_2, ..., \ell_k]$ corresponds to a clustering of all the vertices $\mathcal{C}^t$, and the optimization problem (4) is equivalent to

$$\mathcal{C}^{t+1}(u)^* = \underset{\mathcal{C}(u)\in\{1,2,...,k\}}{\arg\min} \sum_{W_{uv}>0} 1[\mathcal{C}(u) \neq \mathcal{C}^t(v)]W_{uv} - \sum_{W_{uv}<0} 1[\mathcal{C}(u) = \mathcal{C}^t(v)]W_{uv} \ ,$$

where the objective can be treated as a measurement of diversity between two cluster assignments $\mathcal{C}$ and $\mathcal{C}^t$. More specifically, define a function

$$f(\mathcal{C}_1, \mathcal{C}_2) = \sum_u \left[\sum_{W_{uv}>0} 1[\mathcal{C}_1(u) \neq \mathcal{C}_2(v)]W_{uv} - \sum_{W_{uv}<0} 1[\mathcal{C}_1(u) = \mathcal{C}_2(v)]W_{uv} \right] \ ,$$

it is easy to prove that f is non-negative: $f(\mathcal{C}_1, \mathcal{C}_2) \geq 0$ and f is symmetric: $f(\mathcal{C}_1, \mathcal{C}_2) = f(\mathcal{C}_2, \mathcal{C}_1)$. By the end of each iteration, the clustering of all the vertices is updated to

Algorithm 1. Basic Pseudo-EM Routine

Input : Affinity matrix $W^{n\times n}$, initial L^0, `Online`
Output: Cluster assignments of all the vertices

```
t = 0;
repeat
    for u ∈ {1, ..., n} do
        compute optimal C^{t+1}(u) according to (4);
        if Online == True then
            L^t(C^t(u), u) = −1; L^t(C^{t+1}(u), u) = 1;
        end if
    end for
    Update L^{t+1} determined by C^{t+1};
    t = t + 1;
until the partition determined by C does not change
return C
```

$\mathcal{C}^{t+1} = \arg\min_{\mathcal{C}} f(\mathcal{C}, \mathcal{C}^t)$, therefore, we have $0 \leq f(\mathcal{C}^{t+2}, \mathcal{C}^{t+1}) = \min_{\mathcal{C}} f(\mathcal{C}, \mathcal{C}^{t+1}) \leq f(\mathcal{C}^t, \mathcal{C}^{t+1}) = f(\mathcal{C}^{t+1}, \mathcal{C}^t)$, which guarantees that the function value of f or the summation of the objective value in (4) monotonically decreases while being lower-bounded by 0. As a consequence, Algorithm 1 is guaranteed to converge. □

3.2 Discussion

In the above, we are motivated by solving for the cluster label assignments directly, and propose the basic pseudo-EM routine for correlation clustering optimization. However, the proposed algorithm works by starting with an initial L. In this section we will introduce how to initialize L. Apart from that, we will also look into the sparsity issue in data set. At last, we will analyse the computational complexity.

In principle, any L which satisfies (2) is legal. An example is to initialize L with $2I^{n\times n} - 1$. On one hand, it is not difficult to find that the number of clusters decreases along with iterations in our method, therefore one may want to set the number of rows of initial L (i.e. the initial number of clusters) to a relatively large value. On the other hand, as the number of rows of initial L grows, the time complexity of the algorithm increases and the speed of convergence decreases. Here we describe a heuristic initialization of L based on the positive degree of vertices. First, all vertices are sorted in a list by the positive degree in descending order. Starting from the first vertex u in the list, a cluster indicator vector l is constructed by setting $l(v) = 1$ if vertex v has a positive relation with u and is currently in the list, $l(v) = -1$ otherwise, then the vertex u and v that $l(v) = 1$ are removed from the list. These steps are repeated until the list is empty. Then we get a initial L.

However, the sparsity of the affinity matrix leads to ineffectiveness when merging cluster indicator vectors, which will result in a relatively large number of clusters. To address this problem caused by sparse data, we append a sparsity factor of $\mathtt{sum}(L^t_{\mathcal{C}(u),:} + \mathbf{1})/2$ (the size of current cluster $\mathcal{C}(u)$) to the objective to discourage small clusters when optimizing for $\mathcal{C}(u)^*$ in (4). In our experiments when dealing with sparse data, we follow this routine.

Algorithm 2. Initialize L

Input : Affinity matrix $W^{n\times n}$
Output: Cluster indicator matrix L
$L = \emptyset$; $\forall i, j$, $G_{ij} = 2(W_{ij} > 0) - 1$; $\forall i$, $G_{ii} = 1$;
$pdegree =$ **sum**$(G, 2)$; $list =$ **sort**$(pdegree, \texttt{descend})$;
while $list$ is not empty **do**
  $u = list$.**pop**(); $\ell^{1\times n} = -1$;
  for $v \in list$ **do**
    if $G(u, v) == 1$ **then**
      $\ell_v = 1$; $list$.**remove**(v);
    end if
  end for
  $L = L \cup \ell$;
end while
return L

The computational complexity of our algorithm is linear in each iteration as we will see. Let n, a and r denote the number of vertices, the number of attraction edges and the number of repulsion edges respectively. The cost of initializing L is $O(a+nlogn)$, while the proposed algorithm involves complexity of $O(a + r + n)$ in each iteration, which is linear with the sum of the number of edges and the number of vertices. Obviously, it can be seen that the more sparse W is, the less time complexity it will achieve. This is a very useful property to be exploited, especially for large-scale problems in real applications. The number of iterations taken to convergence is relatively small from experience.

A related algorithm to pseudo-EM is the LocalSearch method proposed in the Clustering Aggregation framework [14]. The two algorithms are similar in the sense that they both can be used to optimize correlation clustering. However, the fundamental difference is that LocalSearch is based on a greedy vertex-wise manner, while ours works like EM algorithm.

4 Experiments

In this section, we evaluate the performance of our proposed algorithm pseudo-EM using both synthetic and real data. To investigate the numerical performance, we first conduct comparison with the optimal solution to the correlation clustering problem (1) and the SDP relaxation [6] on toy data. We then compare them to the algorithms including Swap-and-Explore, Expand-and-Explore and Adaptive-label ICM proposed in [2], which aims at large-scale correlation clustering. In synthetic experiments, we also compare with k-means which is the representative of traditional clustering methods. To generate general affinity matrices and the ground truth of clustering, we follow the recipe in [2]. In terms of real data, we conduct experiments on image segmentation and community detection tasks. To evaluate the quality of clustering with the ground truth, we use F_1-measure and recovery levels of k. F_1-measure takes value from $[0, 1]$, and a larger value implies higher quality of clustering; while the recovery level of k is the difference ratio of the selected k to the ground truth, where smaller values indicate better recovery of the number of clusters.

4.1 Numerical Comparison

To investigate the numerical performance of the proposed optimization methods, we randomly generate a problem with 20 vertices and 2 clusters. We first find the exact solution to problem (1) by searching over all possible clustering assignments, which adds up to $\sum_{i=1}^{20} C_{20}^{i} = 1,048,575$ clusterings, and choosing the one with the minimum objective value. We then compare the SDP relaxation and the proposed methods with the optimal solution. SDP is implemented with the semi-definite programming package SeDuMi [15] and the YALMIP toolbox[1]. A subtlety here is that the SDP objective is to maximize agreements, therefore a postprocessing step is taken following the SDP procedure to convert the result to the minimal disagreements solution.

Table 1 summarizes the comparison in terms of the loss objectives given the discretized solutions and time costs of optimization. We can first observe that the SDP relaxation is more effective than solving the problem exactly. However, the loss objective based on the discretized solution of the relaxed continuous result produced by SDP is greater than the exact loss. In the meantime, our proposed methods including pseudo-EM (P-EM) and online pseudo-EM (OP-EM) find solutions with loss objectives closer to the optimal one and significant improvements in time complexity.

4.2 Synthetic Data

To generate synthetic affinity matrix W, n vertices are assigned to c clusters with different sizes (size ratio between the largest and smallest clusters is 100) randomly. For each vertex, we sample the same number of neighbors with nonzero affinities. Then the adjacency matrix of ground truth is corrupted with noise both on the signs and values to get a real affinity matrix.

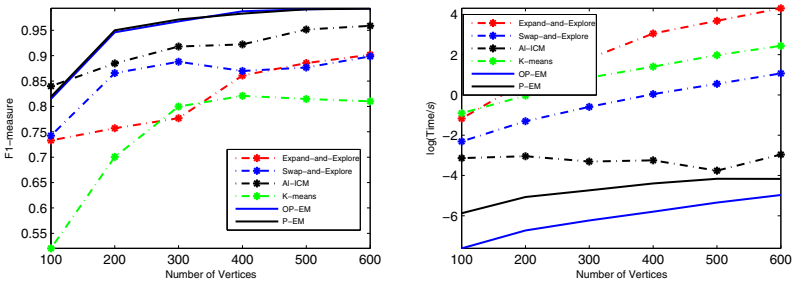
Due to the high computational complexity of Expand-and-Explore, Swap-and-Explore and k-means, we only compare with them as well as adaptive-label ICM on small scale data with the following parameters setting: $\#Clusters = 30$, $\#Neighbors = 20$, $Balance = 0.5$, $Noise = 0.1$, where $Balance$ is the ratio of the number of inter and intra cluster neighbors. The parameter k for k-means is set as the ground truth. Fig. 1 shows the F_1-measure and running time of different algorithms as the number of vertices increases. It can be seen that our methods produce more accurate clustering than the competing algorithms. On the other hand, although being provided with the true number of clusters, the clustering quality produced by k-means is not comparable to the rest of the algorithms. Another observation is that as the number of vertices grows, the clustering quality increases, which makes sense in that one can obtain better knowledge of the underlying distribution given more data. In addition, as mentioned above, the computational costs of Expand-and-Explore, Swap-and-Explore and k-means are high, therefore we will not include them in the following comparison on real data sets.

We further investigate the ability of different algorithms to automatically select the number of clusters k, which is illustrated in Fig. 2. As can be seen, the proposed methods are shown to be significantly more effective at selecting k. Similarly, we can observe that more data makes the estimate of k more accurate; while the difficulty of selecting k increases with the number of clusters given the same amount of data.

[1] `http://users.isy.liu.se/johanl/yalmip/`

Table 1. Loss comparison of exact optimum and SDP/P-EM/OP-EM

	Exact solution	SDP	P-EM	OP-EM
Loss	9.3588	78.2503	35.9834	35.9834
Time/s	858.7878	35.3129	6.1241e-4	5.6880e-4



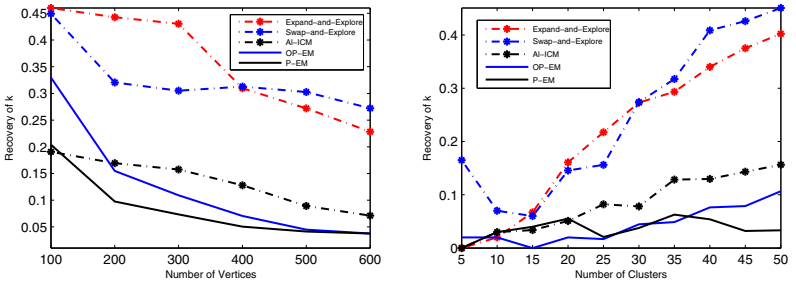
(a) F_1-measure (b) Time

Fig. 1. Small scale data, #Clusters=30, #Neighbors=20, Balance=0.5, Noise=0.1. *(a) F_1-measure. (b) Running time.*

4.3 Image Data

To further demonstrate the scalability and quality of clustering of the proposed algorithms, we conduct experiments on the pixel-level image segmentation task. We take the image data in [16] and rescale the images to 134×200 for illustration. We use the classical normalized cut [17] as comparison.

Before running the algorithms, one should notice that the affinity matrix for correlation clustering consists of both positive and negative entries, which is different from normalized cut. Therefore, to construct the sparse affinity matrix, we first take the affinity matrix computed for normalized cut, followed by a nonlinear transformation to



(a) Effect of number of vertices (b) Effect of number of clusters

Fig. 2. Recovery of k. *(a) #Clusters=30, #Neighbors=20, Balance=0.5, Noise=0.1. (b) #Vertices=500, #Neighbors=20, Balance=0.5, Noise=0.1.*

convert the nonnegative affinities to real valued affinities. The nonlinear transformation function is modified from $y = log\frac{x}{1-x}$ proposed by [1]. To avoid the problem occurred when $x = 1$, here we use the transformation $y = log\frac{1+(x-\delta)}{1-(x-\delta)}$ with the following properties: $y > 0$ when $x > \delta$; $y < 0$ when $x < \delta$ and $y = 0$ when $x = \delta$, where δ is a super parameter and is fixed to 0.05 in our experiments. Another subtlety regarding the comparison is the number of clusters k has to be predefined for normalized cut. In our experiments we set $k = 5$, while our methods automatically select k.

The comparison of image segmentation results is shown in Fig. 3, where the segmentation results of normalized cut are shown in the left column, while the results of our methods including OP-EM and OP-EM-S (OP-EM with sparsity factor) are shown in the middle and right columns respectively. From Fig. 3 we can observe that OP-EM-S is very effective at finding segments on images while OP-EM itself performs not as well, which justifies the idea that adding sparsity factor in (4) could be more effective for sparse data. On the other hand, the left column shows that for normalized cut, inappropriate value of k would result in improper segmentation, which is one of the disadvantages of traditional clustering methods. As comparison, correlation clustering does not need to predefine the k value, which would alleviate the problems caused by unknown number of clusters in the data.

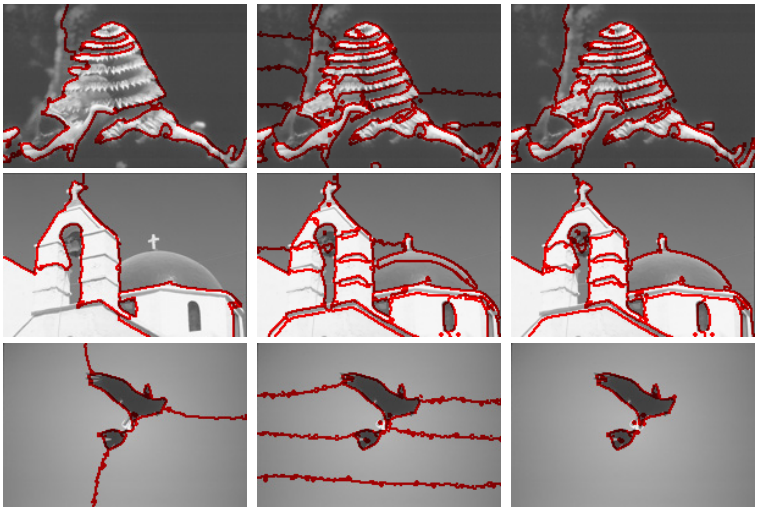


Fig. 3. Image segmentation on Berkeley Segmentation Dataset. *Left to right column: Normalized cut, online pseudo-EM (OP-EM) and online pseudo-EM with sparsity factor (OP-EM-S).*

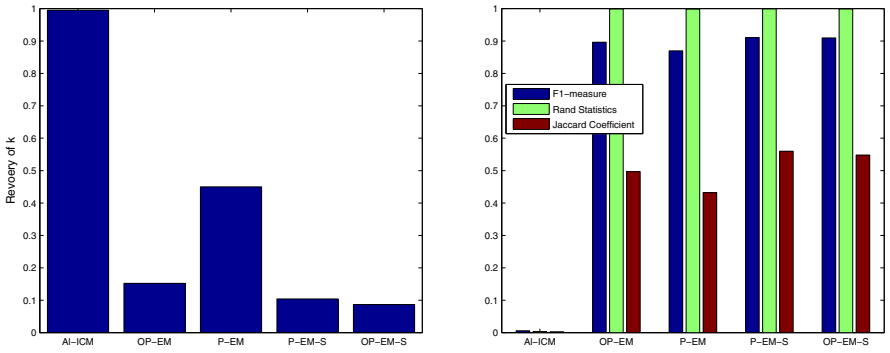
4.4 Social Network Data

Another natural application of clustering is community detection in networks. Here we conduct experiments on Amazon product co-purchasing network with a ground truth distribution of communities[2]. The original network contains 334,863 nodes and 925,872 edges. The number of communities in the network is 151,037 and average community

[2] `http://snap.stanford.edu/data/com-Amazon.html`

size is 19.78. Here we preprocess the data by keeping the top 5,000 communities and removing the redundant clusters and duplicate nodes. As a result we get a network with 16,685 nodes and 1,145 communities for our investigation.

We adopt the commonly used *Jaccard's coefficient* metric to measure the similarity between two nodes in the network [18], then convert the similarities to construct an affinity matrix as described in 4.3. Given the affinity matrix, we apply correlation clustering to detect the communities. We use the following metrics to evaluate the quality of the communities detected: F_1-measure, Rand Statistic and Jaccard Coefficient. All of these measures take value from $[0, 1]$, and larger values imply higher quality of clustering. Fig. 4 summarizes the comparison of the algorithms in terms of recovery of k and quality of clustering, which shows a clear advantage of our algorithms.



(a) Recovery of k (b) Clustering evaluation

Fig. 4. Community detection on Amazon network data

5 Conclusion

In this paper we propose algorithms for solving general correlation clustering which could handle large-scale data from a different perspective by decoupling the affinity matrix and the cluster indicator matrix followed by a pseudo-EM optimization. To further improve the quality of optimization and alleviate the problem of local minimum, we adopt online udpates and append a sparsity factor for sparse data. Experimental results on both synthetic and real data demonstrate the effectiveness of the proposed techniques.

Correlation clustering is based on graphs with two types of edges: positive and negative, while in many real problems such as social networks or protein-protein interaction networks, the types of edges can be more diverse. Therefore, an interesting direction for future work will be to further generalize the proposed methodology to graphs with more diverse relations or interactions. Another important problem to look into is the theoretical analysis of the optimization quality of the proposed methods.

Acknowledgments. Research supported by the National Natural Science Foundation of China (No. 61003135) and the Fundamental Research Funds for the Central Universities (WK0110000022, WK0110000036).

References

1. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. In: Proceedings of the 43rd Symposium on Foundations of Computer Science, FOCS 2002, p. 238. IEEE Computer Society, Washington, DC (2002)
2. Bagon, S., Galun, M.: Large scale correlation clustering optimization. CoRR abs/1112.2903 (2011)
3. Joachims, T., Hopcroft, J.: Error bounds for correlation clustering. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 385–392. ACM, New York (2005)
4. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pp. 524–533 (2003)
5. Demaine, E.D., Immorlica, N.: Correlation clustering with partial information. In: Arora, S., Jansen, K., Rolim, J.D.P., Sahai, A. (eds.) RANDOM 2003 and APPROX 2003. LNCS, vol. 2764, pp. 1–13. Springer, Heidelberg (2003)
6. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, pp. 526–527. Society for Industrial and Applied Mathematics, Philadelphia (2004)
7. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. J. ACM 55(5), 23:1–23:27 (2008)
8. Nowozin, S., Jegelka, S.: Solution stability in linear programming relaxations: graph partitioning and unsupervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 769–776. ACM, New York (2009)
9. Glasner, D., Vitaladevuni, S.N., Basri, R.: Contour-based joint clustering of multiple segmentations. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, pp. 2385–2392. IEEE Computer Society, Washington, DC (2011)
10. Vitaladevuni, S., Basri, R.: Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2203–2210 (2010)
11. Yurii, N., Arkadii, N.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics (1994)
12. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
13. Besag, J.: On the Statistical Analysis of Dirty Pictures. Journal of the Royal Statistical Society, Series B (Methodological) 48(3), 259–302 (1986)
14. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Trans. Knowl. Discov. Data 1(1) (March 2007)
15. Sturm, J.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optimization Methods and Software 11-12 (1999)
16. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 898–916 (2011)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
18. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of the Twelfth International Conference on Information and Knowledge Management, CIKM 2003, pp. 556–559. ACM, New York (2003)

A Fast Spectral Clustering Method Based on Growing Vector Quantization for Large Data Sets*

Xiujun Wang[1,2], Xiao Zheng[1], Feng Qin[1], and Baohua Zhao[2]

[1] School of Computer Science and Technology, Anhui University of Technology, China
[2] School of Computer Science and Technology, University of Science and Technology of China
wxj@mail.ustc.edu.cn, zhx@ahut.edu.cn

Abstract. Spectral clustering is a flexible clustering algorithm that can produce high-quality clusters on small scale data sets, but it is limited applicable to large scale data sets because it needs $O(n^3)$ computational operations to process a data set of n data points[1]. Based on the minimization of the increment of distortion, we tackle this problem by developing a novel efficient growing vector quantization method to preprocess a large scale data set, which can compress the original data set into a small set of representative data points in one scan of the original data set. Then we apply spectral clustering algorithm to the small set. Experiments on real data sets show that our method provides fast and accurate clustering results.

Keywords: Spectral clustering, growing vector quantization, distortion.

1 Introduction

Clustering is an important problem in machine learning and data mining [1][8][9] [10][14][15][16][17][18][19][20]. A large number of algorithms have been proposed for clustering problems. Recently, spectral clustering has become more and more popular for its high-accurate clustering advantage over traditional clustering algorithms. Spectral clustering methods is a class of methods based on the theories of eigen-decompositions of affinity matrix and kernel matrix, and there has been many results on the theoretical basis of spectral clustering[14][18]. Many traditional clustering algorithms strongly depend on Euclidean space, and can only detect cluster of convex geometry, in contrast, spectral clustering algorithms are more flexible, and able to detect clusters of a large number of geometries. Spectral clustering methods often provide superior empirical clustering results when compared with traditional clustering algorithms, such as k-means, and they have been applied in a large number real world applications in bioinformatics, robotics and so on.

* This work is supported by National Natural Science Foundation of China under Grant No. 61003311, Jiangsu Provincial Key Laboratory of Network and Information Security Grant No. BM2003201-201006, Anhui Provincial Natural Science Research Key Project of China under Grant No. KJ2011A040, Youth Foundation of Anhui University of Technology under Grant No. QZ201316.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 25–33, 2013.

Despite the advantages of spectral clustering, it can not process the date set with a large number of data points efficiently. The reason is pointed in [1]: given a date set with n data points, spectral clustering algorithms always need $O(n^3)$ computational operations to construct a $n \times n$ affinity matrix of data points and compute eigenvalues and eigenvectors of the affinity matrix. When the data point number exceeds the order of thousands, spectral clustering algorithms needs more than $O(10^9)$ computational operations and thus can not be efficient and feasible to real world applications.

To tackle the above problems, there are mainly four kinds of solutions. One is based on replacing the original data set with a small number of representative points (these data points aim to preserve the original clustering-relevant structure in the original data set), then it clusters the data set of representative points and uses the clustering result as the final clustering results[1]. The second is based on subsampling on the original data set then replaces the original set with the subsample set, it then clusters the subsample set as the final clustering results[8]. The third focuses on using a low-rank matrix to approximate the affinity matrix of the original data set, then do spectral clustering based on the constructed low-rank matrix[8][9][10]. It has been pointed in [1] that all of these three solutions neglect the important connection between the decreasing of the original data set by a preprocessing method and the subsequent effect on the clustering. In [1], Yan et. al. proposed two fast approximate spectral clustering algorithms based on rate-distortion theory. The first method in [1] is a local k-means clustering algorithm (KASP) (for the information theoretical aspect of k-means algorithm, please refer to [20]). It needs a user specific parameter k (representative point number in k-means for KASP), which is hard for common user to specify. KASP needs $O(nkt)$ computational operations to preprocess the original data and produce a set of k representative data points, which requires t times scanning of the original data, this can be a heavy burden when the preprocessing method in KASP needs a large number of iteration to converge. The second method in [1] is a fast spectral clustering algorithm with RP trees(random projection tree) also needs a user specific parameter k and $O(hn)$ to construct the h-level projection tree[13]. But to ensure a small distortion with high probability, h always needs to be $O(d \log d)$ where d is dimensionality of the original data set. The second method in [1] is slower and less accurate than the first according to the experiments in [1].

There are also some interesting and important works in the improvement of the spectral clustering algorithm based on the change of the original optimization problem in spectral clustering. For example graph-based relaxed clustering (GRC) is proposed in [18]. Qian et. al. proposed a fast graph-based relaxed clustering algorithm(FGRC) in [14], which uses core-set-based minimal enclosing ball approximation. FGRC improves the sensitiveness to the adopted similarity measure and reduces time complexity.

In this paper, we propose a novel growing vector quantization method (GVQ) to preprocess the original data set, which provides a set of representative data point based on the minimization of the increment of distortion. Then we cluster the new generated set of representative data points as the final clustering results with the spectral clustering algorithm in [2]. GVQ requires user-input parameters θ: maximum

distortion of a representative data point and γ: maximal distance. It can produce a small set of representative data point in one-scan of the original data set, and requires $O(n)$ computational operations. Experimental results over real data sets show that our method achieves a similar clustering accuracy and consuming much less computational time, compared with KASP[1].

The rest of the paper is organized as follows. In Section 2, we introduce some background knowledge in vector quantization and spectral clustering. GVQ method will be introduced in section 3. Experimental results and conclusion will be reported in section 4 and 5.

2 Preliminaries

2.1 Spectral Clustering

The spectral clustering algorithm in [2] is one of the most popular spectral clustering algorithms. It introduces a special processing manner of using the first k eigenvectors of the affinity matrix and provides conditions under which the spectral clustering in [2] will perform well. The algorithm will be listed as follows[2]. We adopt a similar notation used in [2].

Algorithm 1. Spectral clustering
Input: number of clusters k, affinity matrix $W \in R^{n\times n}$ of data set $\{x_1,..,x_n\}$

Step1. Compute laplacian matrix $L = D - W$, D is a diagonal matrix with element $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$

Step2. Compute the first k eigenvectors $u_1,..,u_k$ of the generalized eigenproblem $Lu = \lambda Du$; let $Z = [u_1,..,u_k]^T$

Step3. Cluster the row vector $(y_i), i = 1..n$ in Z with k-means algorithm into clusters $C_1,..,C_k$

Output: k clusters: $A_1,..,A_k$ $A_i = \{j \mid y_j \in C_i\}$

The total computational operations in spectral cluster is $O(n^3)$.

2.2 Vector Quantization

Vector quantization [3][11][12][13] is a process that encodes each input vector (a data point) with the closest matching (with minimum distance) vector in current codebook (representative data point set), then decodes each vector according the generated codebook. The crucial part of this method is to design a good codebook (a set of representative data point set). The well-known algorithm for this design is

Linde-Buzo-Gray algorithm. The LBG algorithm can provide a encoder that satisfy necessary but not sufficient conditions for optimality. But it needs fixed input number of representative data point and iterated scans of the original data set until the representative data points in the codebook change by a small enough fraction compared with the latest iteration. Given n original data points, k representative data points in codebook and t iterations, LBG needs computational operations O(knt) to construct a codebook. There also some sophisticated variants of LBG, in which a number of representative data points is updated to encode a new input vector, which usually have a larger computational operations than LBG [4].

Thus it is time-consuming to applying the traditional vector quantization algorithm directly into the preprocessing of a large set of data point.

3 Our Method

When preprocessing a large data set and generating a good set of representative data point to replace the original set, there are two important factors. The first is: the distortion of the preprocessing algorithm, which is crucial for the following clusters quality[1]. The second is: the scan number of all data points in the original data set, given that many scans of a larger data set is time-consuming.

Based on the above considerations, we design a novel growing vector quantization method GVQ based on the minimization of the increment of distortion, which can produce a good set of representative data point in one-scan of the original data set.

We will first prove that the traditional LBG algorithm is not optimal, when we consider the increment of distortion for encode each newly input vector. Then we will give GVQ algorithm.

3.1 Minimization of the Increment of Distortion

Let us assume a representative data point $r \in R = \{r_1,..,r_k\}$ in a codebook and r has encoded some points $\text{set}_r = \{y_1,..,y_s\} \subset X$ of the original data $X = \{x_1,..,x_n\}$. We define the distortion of r to encode set_r as: $\text{D}(r)=\sum_{i=1}^{s} \| r - y_i \|^2$ ($\| \bullet \|$ is Euclidean measure and $y_i \in X$). It is easy to see that when $r = (1/s)\sum_{i=1}^{s} y_i$, $\text{D}(r)$ is minimized, and we call this r best represents its encoded set_r. In the following we assume that each representative data point in R always best represents its encoded data after processing each data point of the original data sets X.

Assume that LBG has processed the first $m-1$ data points of X and it is about to encode x_m, and the closest matching representative data point in the current codebook $R = \{r_1,..,r_k\}$ is r_1, which means that $\| r_1 - x_m \|^2 < \| r_i - x_m \|^2, r_i \in R, r_i \neq r_1$. We also assume that r_1 has encoded $\text{Set}_{r_1} = \{z_1,..,z_s\}\ z_i \in \{x_1,..x_{m-1}\}, i = 1..s$. Then LBG algorithm will select r_1 to encode x_m. The increment of distortion is:

$$\Delta E = \sum_{i=1}^{s} \| r_1' - z_i \|^2 + \| r_1' - x_m \|^2 - \sum_{i=1}^{s} \| r_1 - z_i \|^2 \quad (1)$$

$$r_1 = (1/s) \sum_{i=1}^{s} z_i \quad (2)$$

$$r_1' = \sum_{i=1}^{s} z_i / (s+1) + x_m / (s+1) \quad (3)$$

After some reduction of (1) (2) (3) we get:

$$\Delta E = s \| r_1 - x_m \|^2 / (s+1) \quad (4)$$

From (4), we can see that the increment of distortion depends not only on the distance between the representative data point (before encode x_m) and x_m, but also the number of original data points r_1 has encoded. Thus for each data point of X that about to be encoded, the choice made in LBG is not optimal according to distortion increment, because it only consider the distance between representative data points and an uncoded data point.

3.2 Growing Vector Quantization Method

It is hard for a traditional vector quantization algorithm to assign the number of representative data points of a codebook before processing a set of original data points, which is due to the lack of knowledge about the distributional characteristics of the original data set. We also note the a predefined and fixed number of representative data points is hard for common user to specify, which also troubles the understanding of final clustering results.

Algorithm 2. Growing vector quantization
Input: maximal distance γ, maximal distortion θ, data set $X = \{x_1, .., x_n\}$, a empty set R of representative data points.
For each $x_i \in X, i = 1..n$

Step1. Choose the closest matching representative data point $r \in R$ according to (4), and $D(r) < \theta$. If the representative data point with minimal distortion has distortion larger than θ, we choose the representative data point with the second minimal distortion and so on.

Step2.If r cannot be chosen in step 1 or the chosen r has $\| r - x_i \| > \gamma$, add a new representative data point r_{new} to R, let $r_{\text{new}} = x_i$, $D(r_{\text{new}}) = 0$ and go to process x_{i+1}

Step3. Suppose r has encoded s data points in X before encodes x_i, update r and $D(r)$: $r = (r \times s + x_i)/(s+1)$, $D(r) = s \| r - x_i \|^2 / (s+1)$ and go to process x_{i+1}

Given maximal distance γ and maximal distortion θ, we design a growing mechanism in the following two steps:

Step1. Given an uncoded data point x in X (has not been encoded by representative data points in codebook), we choose the closest matching representative data point r according to the minimization of formula (4) and $\mathrm{D}(r) < \theta$. That is: if the representative data point with the minimal value of (4) has a distortion value smaller a predefined value θ, then we choose it as r, if not, we test if the representative data point with the second minimal value of (4) has distortion larger than θ, and so on.

Step2. If we cannot find a $r \in R$ or we find a $r \in R$ in step1 has $\| r - x \|^2$ larger than a predefined value γ, then we adds a new representative data point r_{new} into R, and encode x with r_{new}. Otherwise, we use $r \in R$ chosen in step 1 to encode x.

Thus our method will add a new representative data point r_{new} into R (the codebook) when we can't find a $r \in R$ in step 1 or $r \in R$ chosen in step 1 has $\| r - x \|^2 > \gamma$.

It should be noted that θ and γ are closely related to representative data set distortion, thus gives a better interpretation for the final clustering results than predefined representative data point number in [1].

With the growing mechanism and (4), the proposed GVQ algorithms is as followed:

Theorem 1. Given maximal distance γ, maximal distortion θ and a data set $X = \{x_1, .., x_n\}$, growing vector quantization algorithm will produce a representative data set R with at most $n\gamma / \theta$ data points approximately, when γ is set to be larger than the maximal distance of data point pairs in X and growing vector quantization algorithm has computational operations at most $\mathrm{O}(n^2 \gamma / \theta)$.

Proof. When γ is set to be larger than the maximal distance of the data point pairs, then for each uncoded data point in X, growing vector quantization algorithm will add a new representative data point to R if R is empty or the distortions of representative data points in $r \in R$ all exceeds θ. It means that new representative data point will be added when all $r \in R$ has $\mathrm{D}(r) > \theta$.

Given a $r \in R$ and maximal distortion θ, the minimal number of data points in X that can be encoded by r is θ / γ, thus the n data points in X can be encoded by at most $n\gamma / \theta$ representative data points of R.

It is obvious that growing vector quantization algorithm(GVQ) make one scan of X and for each $x \in X$, GVQ needs $|R|$ computation operations to compute the distortion increment (4) for each $r \in R$. Thus the total computational operations is $\mathrm{O}(n^2 \gamma / \theta)$. ■

It should noted that after GVQ generated R, we use the spectral algorithm in Algorithm 1 to cluster R, then output its clusters as the final clustering results for X. By Theorem 1, it is obvious that the total computational operations for clustering a

data set $X = \{x_1, .., x_n\}$ with GVQ and Algorithm 1 (denoted by GVQ+Spectral clustering) needs total computational operations $O(n^2\gamma/\theta + n^3\gamma^3/\theta^3)$ at most when we set a large γ. Give the same reduction ratio λ (in GVQ $\lambda = \theta/\gamma$ approximately), which is the ratio between the number of the generated representative data points and the number of data points in original set X , KASP in [1] always needs computational cost $t \times n^2/\lambda + n^3/\lambda^3$ (t usually is not a small constant in order to make k-means converge), where our method (GVQ+Spectral clustering) needs computational cost $n^2/\lambda + n^3/\lambda^3$ at most.

4 Experiment

4.1 Data Sets

RCV1(Reuters Corpus Volume I)[5] contains 781,256 documents which are categorized into 350 classes. We remove those categories which contains less 500 documents or is multi-labeled, and select a subset of about 20,000 documents in the remaining 103 categories as the testing set. In this experiment, we test our method, KASP in [1] and spectral clustering method [2] respectively on 30 categories, 60 categories and 90 categories of the remaining 103 categories. The 30 categories has the largest value of average sample number per category, while the 60 categories has the second largest value.

4.2 Evaluation Metrics

We test the three methods: GVQ+Spectral clustering, KASP[1], spectral clustering[2] by computing Clustering Accuracy (CA) and Normalized Mutual Information (NMI) on the labels generated by the three methods and their real labels. Clustering accuracy and Normalized information have been defined and used in [1][2][6]. It should be noted that for the spectral clustering algorithm used in these three method, we choose to construct 20-nearest neighbor graph and gaussian similarity function with variance calculated in a self-tuning way suggested by [7].

We also use the clustering time as an important clustering performance quantity.

Our experiments were performed on a Window 7 machine with 2.0GHz Cpu and 4 GB main memory. All methods were implemented in Matlab.

4.3 Results

Fig.1 , Fig.2 and Fig. 3 shows that NMI and CA values of the three method over 30,60,90 categories of RVC1. It is easy to see that both NMI and CA of GVQ+spectral clustering are similar to KASP. These two methods both have lower NMI and CA values than spectral clustering. This is because that the distortion incurred by the processing of GVQ and KASP. But Fig. 4 shows that both KASP and our method run much faster than spectral clustering and our method is faster than KASP.

5 Conclusion

A fast spectral clustering algorithm for large data set is proposed in this paper. Based on the minimization of the increment of distortion, we develop a novel efficient growing vector quantization method to preprocess a large scale data set, which can compresses the original data set into a small set of representative data points in one scan of the original data set. Then we apply spectral clustering algorithm to the small set. Experiments on real data sets show that our method provides fast and accurate clustering results.

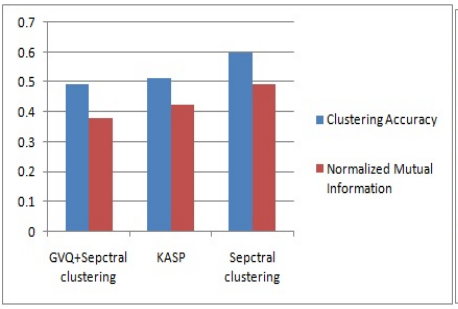


Fig. 1. CA and NMI on 30 categories

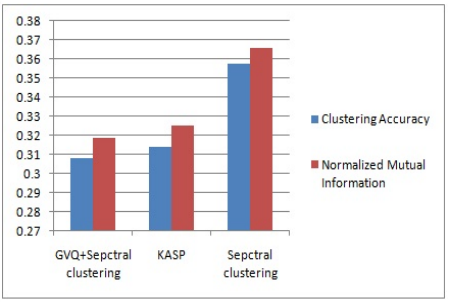


Fig. 2. CA and NMI on 60 categories

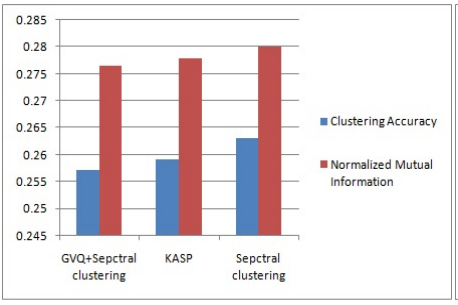


Fig. 3. CA and NMI on 90 categories

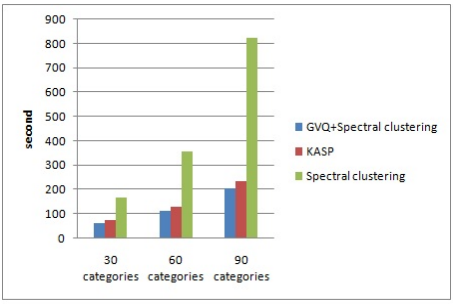


Fig. 4. Running time over 30, 60, 90 categories of GVQ+Spectral clustering, KASP and Spectral clustering

References

1. Yan, D., Huang, L., Jordan, M.I.: Fast approximate spectral clustering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 907–916. ACM (2009)
2. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Proceedings of the Advances in Neural Information Processing Systems, pp. 849–856. MIT Press, Cambridge (2002)

3. Equitz, W.H.: A new vector quantization clustering algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing 37(10), 1568–1575 (1989)
4. Vasuki, A., Vanathi, P.T.: A review of vector quantization techniques. IEEE Potentials 25(4), 39–47 (2006)
5. Lewis, D.D., Yang, Y., Rose, T.G., et al.: Rcv1: A new benchmark collection for text categorization research. The Journal of Machine Learning Research 5, 361–397 (2004)
6. Chen, W.Y., Song, Y., Bai, H., et al.: Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3), 568–586 (2011)
7. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems, pp. 1601–1608 (2004)
8. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
9. Wauthier, F.L., Jojic, N., Jordan, M.I.: Active spectral clustering via iterative uncertainty reduction. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1339–1347. ACM (2012)
10. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 614–621. IEEE (2012)
11. Pagès, G., Wilbertz, B.: Intrinsic stationarity for vector quantization: Foundation of dual quantization. SIAM Journal on Numerical Analysis 50(2), 747–780 (2012)
12. Kästner, M., Hammer, B., Biehl, M., et al.: Functional relevance learning in generalized learning vector quantization. Neurocomputing 90, 85–95 (2012)
13. Dasgupta, S., Freund, Y.: Random projection trees for vector quantization. IEEE Transactions on Information Theory 55(7), 3229–3242 (2009)
14. Qian, P., Chung, F.L., Wang, S., et al.: Fast Graph-Based Relaxed Clustering for Large Data Sets Using Minimal Enclosing Ball. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42(3), 672–687 (2012)
15. Deng, Z., Choi, K.S., Chung, F.L., et al.: Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recognition 43(3), 767–781 (2010)
16. Wang, S., Chung, K.F.-L., Deng, Z., Hu, D., Wu, X.: Robust maximum entropy clustering algorithm with its labeling for outliers. Soft Comput. 10(7), 555–563 (2006)
17. Wang, S., Chung, K.F.-L., Deng, Z., Hu, D.: Robust fuzzy clustering neural network based on epsilon-insensitive loss function. Appl. Soft Comput. 7(2), 577–584 (2007)
18. Lee, C.H., Zaïane, O.R., Park, H.H., et al.: Clustering high dimensional data: A graph-based relaxed optimization approach. Information Sciences 178(23), 4501–4511 (2008)
19. Cao, J., Wu, Z., Wu, J., et al.: SAIL: Summation-bAsed Incremental Learning for Information-Theoretic Text Clustering (2013)
20. Cao, J., Wu, Z., Wu, J., et al.: Towards information-theoretic K-means clustering for image indexing. Signal Processing (2012)

A Novel Deterministic Sampling Technique to Speedup Clustering Algorithms

Sanguthevar Rajasekaran and Subrata Saha

Department of Computer Science and Engineering
University of Connecticut, Storrs, USA
{rajasek,subrata.saha}@engr.uconn.edu

Abstract. Conventional clustering algorithms suffer from poor scalability, especially when the data dimension is very large. It may take even days to cluster large datasets. For applications such as weather forecasting, time plays a crucial role and such run times are unacceptable. It is perfectly relevant to get even approximate clusters if we can do so within a short period of time. In this paper we propose a novel deterministic sampling technique that can be used to speed up any clustering algorithm. We call this technique *DSC (Deterministic Sampling-based Clustering)*. As a case study we consider hierarchical clustering. Our empirical results show that DSC results in a speedup of more than an order of magnitude over exact hierarchical clustering algorithms when the data size is more than 6,000. Also, the accuracy obtained is excellent. In fact, on many datasets, **we get an accuracy that is better than that of exact hierarchical clustering algorithms!**. Even though we demonstrate the power of DSC only with respect to hierarchical clustering, DSC is a generic technique and can be employed in the context of any other clustering technique (such as k-means, k-medians, etc.) as well.

Keywords: Clustering algorithms, Agglomerative hierarchical clustering, Center of gravity, Clustering efficiency.

1 Introduction

The problem of clustering is to partition a given set of objects into groups (called *clusters*) such that objects in the same group are "similar" to each other. There are numerous ways of defining "similarity" and hence there exist many different versions of the clustering problem. Examples include hierarchical clustering, k-means clustering, k-medians clustering, etc. For each of these versions several efficient algorithms have been proposed in the literature. For example, the best known algorithm for hierarchical clustering takes $O(n^2)$ time on n objects (or points). We live in an era of data explosion and the value of n is typically very large. As a result, even a quadratic time algorithm may not be feasible in practice when the datasets are very large. Another factor that could add to the complexity of clustering is the data dimension. One possible way of speeding up these algorithms is with the employment of sampling. For instance, the CURE

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 34–46, 2013.

algorithm [7] employs random sampling in clustering and the speedups obtained are very good. In this paper we propose a novel deterministic sampling technique that can be used to speedup any clustering algorithm. To the best of our knowledge, deterministic sampling has not been utilized before in clustering.

Let I be a given set of n objects. It helps to assume that the objects are points in $\Re^d$ for some large d. In our technique there are several levels of deterministic sampling. In the first level the input is partitioned into several parts. Each part is clustered. Here one could employ any clustering algorithm. Some number of representatives (i.e., sample points) are chosen from each cluster of each part. These representatives move to the next level as a deterministic sample. In general, at each level we have representatives coming from the prior level. These representatives are put together, partitioned, each part is clustered, and representatives from the clusters proceed to the next level. This process continues until the number of points (i.e., sample from the prior level) is 'small' enough. When this happens, these points are clustered into k clusters, where k is the target number of clusters. For each of these clusters, a center is identified. Finally, for each input point, we identify the closest center and this point is assigned to the corresponding cluster.

Clearly, the above technique can be employed in conjunction with any clustering algorithm. In this paper we consider hierarchical clustering as a case study. However, the technique is generic. We have tested our technique on many synthetic as well standard benchmark datasets. We achieve a speedup of more than an order of magnitude over exact hierarchical clustering when the data size is more than 6,000. Please note that real-life datasets have millions of points and more. Also, the accuracy obtained is very impressive. In fact, on many datasets, our accuracy is better than that of exact hierarchical clustering algorithms!

The rest of this paper is organized as follows: Section 2 has a literature survey. Some preliminaries on hierarchical clustering, sampling, clustering accuracy/efficiency, etc. are presented in Section 3. Section 4 describes the proposed algorithm. Analyses of time complexity and accuracy of our algorithm are presented in Section 5. Our experimental platform is explained in Section 6. Section 7 shows our experimental results and Section 8 concludes the paper.

2 Related Works

In this section (due to space constraints) we provide a very brief literature survey. Clustering algorithms fall under different categories such as partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. Partitioning based clustering divides the dataset into some user specified number of clusters using centroid or medoid based procedures. In the centroid based algorithms clusters are formed using the center of gravity of the input points. Medoid based algorithms produce clusters accumulating points closest to the center of gravity. Some notable examples of centroid and medoid based algorithms can be found in [11], [10], [4], [13], and [22]. A hierarchical clustering algorithm partitions the entire dataset into a tree, known

as the *dendrogram*, of clusters. Two kinds of hierarchical clustering, namely, agglomerative and divisive are known. Agglomerative methods form the clusters in a bottom-up fashion where each data point starts as a single cluster and these clusters get progressively merged until all the input points form a single cluster. Divisive approach works in a top-down fashion starting with a single cluster containing all the input points. This cluster gets partitioned into smaller and smaller clusters until each input point forms a single cluster. Some of the well known hierarchical clustering algorithms are: Balanced Iterative Reducing and Clustering using Hierarchies BIRCH [23], Clustering Using REpresentatives CURE [7] and CHAMELEON [12]. As mentioned before, CURE employs random sampling.

Density based clustering partitions the dataset into a number of groups based on the density of the input points in a region. Examples include DBSCAN [6] and DENCLUE [9]. Grid-based clustering algorithms run in two steps. In the first step they map the entire dataset into a finite number of hyper-rectangular cells and in the next step they perform some statistical procedures on the transformed or mapped points to find the density of the cells. Adjacent cells are connected to form a single cluster if those cells follow same density distribution. Example density-based clustering algorithms are STatistical INformation Grid-based method STING [21], WaveCluster [18], and CLustering In QUEst CLIQUE [1].

The above algorithms do not assume any hypothesis/model about the data and fall under the category of *exploratory* algorithms. *Confirmatory* or *inferential* algorithms assume a hypothesis/model on the data to be clustered. A number of statistical inferential techniques can be found in the literature. Examples include linear regression, discriminant analysis, multi-dimensional scaling, factor analysis, principal component analysis, and so on. A survey on inferential clustering can be found in [20]. Some other interesting clustering techniques and algorithms can be found in [3], [14], [2], [19], and [17].

3 Background Information

3.1 Agglomerative Hierarchical Clustering

Steps involved in any agglomerative clustering procedure are shown in Algorithm 1. The distance between two clusters can be defined in a number of ways and accordingly different versions of the hierarchical clustering problem can be obtained. We define below some of these distances. For any two clusters I and J, let $d(I, J)$ stands for the distance between I and J. In step 3 of Algorithm 1, let the clusters with the minimum distance be I and J. Also, let the merged cluster in step 4 be Q. For any cluster I, $|I|$ denotes the size of the cluster I. In the following definitions, L refers to any cluster other than I, J, and Q.

1. Single-link: $d(Q, L) = \min\{d(I, L), d(J, L)\}$. The distance between two clusters A and B is the closest distance between a point in A and a point in B: $d(A, B) = \min_{a \in A, b \in B} d(a, b)$.
2. Complete-link: $d(Q, L) = \max\{d(I, L), d(J, L)\}$. The distance between two clusters A and B is the maximal distance between a point in A and a point in B: $d(A, B) = \max_{a \in A, b \in B} d(a, b)$.

Algorithm 1. Agglomerative Hierarchical Clustering

Input: A set of n data points and an integer k
Output: The best k clusters
begin

1. Start with n clusters (nodes) labeled $1, 2, 3, \ldots, n$, where each cluster has one input point.
2. Calculate all pair-wise cluster distances and place them in a $n \times n$ matrix. This matrix is called the dissimilarity matrix.
3. Find the pair of nodes (i.e., clusters) with the minimum cluster distance.
4. Join these two nodes into a new node and remove the two old nodes. Relabel the nodes with consecutive integers.
5. Update the dissimilarity matrix.
6. Repeat steps 3 through 5 until only k clusters are left. Output these k clusters.

3. Average-link: $d(Q, L) = \frac{|I| \cdot d(I,L) + |J| \cdot d(J,L)}{|I| + |J|}$. The distance between two clusters A and B is the average distance between a point in A and a point in B: $d(A, B) = \frac{1}{|A||B|} \sum_{a \in A, b \in B} d(a, b)$.

4. Centroid-link: $d(K, L) = \frac{|I| \cdot d(I,L) + |J| \cdot d(J,L)}{|I| + |J|} - \frac{|I| \cdot |J| \cdot |d(I,J)|}{(|I| + |J|)^2}$. Here $d(A, B)$ is the distance between the centroids of the clusters in Euclidean space: $d(A, B) = (\| \boldsymbol{c}_A - \boldsymbol{c}_B \|)^2$, where $\boldsymbol{c}_A$ denotes the centroid of the points in cluster A.

5. Ward-link: $d(K, L) = \frac{(|I| + |L|) \cdot d(I,L) + (|J| + |L|) \cdot d(J,L) - |L| \cdot d(I,J)}{|I| + |J| + |L|}$. Here $d(A, B) = \frac{2|A||B|}{|A| + |B|} \cdot (\| \boldsymbol{c}_A - \boldsymbol{c}_B \|)^2$ where $\boldsymbol{c}_A$ denotes the centroid of the points in cluster A.

3.2 Sampling

The idea of sampling is to pick a subset of the given input, process the subset, and make inferences on the original dataset. Sampling has played a major role in the design of efficient algorithms for numerous fundamental problems in computing such as sorting, selection, convex hull, clustering, rules mining, etc. Both sequential and parallel algorithms have benefited. Random sampling perhaps is the most popular. Deterministic sampling has also been employed for such problems as selection. For a survey on the role of random sampling in knowledge discovery, the reader is referred to [15]. In agglomerative hierarchical clustering, random sampling is exploited in ROCK [8] and CURE [7]. These algorithms randomly choose a subset of the input points and cluster this subset. Each of the other input points is assigned to the cluster closest to it.

To the best of our knowledge, deterministic sampling has not been employed in the context of clustering before. A major advantage of deterministic sampling over randomized sampling lies in the fact that the analyses done using deterministic sampling always hold. In this paper we propose a technique called DSC

(Deterministic Sampling-based Clustering). DSC is based on the scheme proposed in [16]. The scheme of [16] works in the context of out-of-core selection. The problem of selection is to find the i^{th} smallest key from a collection X of n keys. The selection algorithm of [16] works as follows: At the beginning all the keys are considered as live keys. The algorithm then goes through stages of sampling. In the first stage, it divides the collection X into a number of parts such that each part contains M keys, where M is the size of the memory. Each part is then sorted and keys that are at a distance of $\sqrt{M}$ from each other are retained. So, the ranks of the retained keys are $(\sqrt{M}, 2\sqrt{M}, 3\sqrt{M}, \ldots)$. Clearly, the number of keys in the retained set R_1 from the first stage is $= \frac{n}{\sqrt{M}}$. In the next stage, the algorithm again groups the elements of R_1 such that there are M elements in each part, sorts each part, collects only every $\sqrt{M}^{th}$ element in each part. Let the set retained in the second stage be R_2. This process of selecting a subset from one level as a sample to the next level continues until only $\leq M$ elements are left. These elements constitute a deterministic sample from which two elements ℓ_1 and ℓ_2 are picked such that these elements bracket the i^{th} smallest element of X. Followed by this, we eliminate all the keys of X that do not have a value in the interval $[\ell_1, \ell_2]$. This process of sampling and elimination is continued until the number of keys left is small. At that point, the remaining elements are sorted and the element of interest is identified.

3.3 Clustering Accuracy

Given a clustering algorithm, there are multiple ways to measure the accuracy of clustering. In this paper we use a measure that is very intuitive and has been mentioned in many prior works (see e.g., [7]). Given k clusters corresponding to a given input point set, we first identify the center of each cluster. Then we calculate the distance of each point to the center of the cluster it belongs to. This distance is summed over all the points. If d_{ij} is the distance of the i^{th} point in cluster j to the corresponding center, the clustering accuracy is computed as $\sum_i \sum_j d_{ij}$. Let the clustering accuracy of DSC and any other exact hierarchical clustering algorithm be CA_A and CA_E, respectively. Then, we define the clustering efficiency of our algorithm as $\frac{CA_E}{CA_A} \times 100\%$.

4 Our Algorithm

There are several levels of sampling in our technique. The number of points that move from one level as a sample to the next level progressively decreases. When the number of points in some level falls below some threshold for the first time, we cluster those points into k clusters. We identify the centers of these clusters. Each input point p is then assigned to the cluster whose center is closest to the point p.

Let the number of levels in the algorithm be r (i.e., in stage r the number of remaining points falls below a threshold for the first time). In the first stage we have all the n input points. We partition the input set into p_1 parts of equal size.

Each part (of size $\frac{n}{p_1}$) is clustered into q clusters using any clustering algorithm. We pick ℓ representatives from each such cluster and these representatives from each cluster of each part move to the second level as a sample. The number of points that move to the second level is $\frac{n}{p_1}q\ell$. In the second stage all of these points are put together, partitioned into p_2 parts of equal size, each part is clustered into q clusters, ℓ representatives are chosen from each cluster, and these representatives move to the third level; and so on.

In general, in level i we partition the points into p_i equal parts, cluster each part into q clusters, pick ℓ points from each cluster, and the picked points move to level $i+1$, for $1 \leq i \leq (r-1)$. A pseudocode for DSC is supplied in Algorithm 2. In this pseudocode we have assumed (for simplicity) that $p_1 = p_2 = \cdots = p_r$. Also, the parameters q and ℓ have to be chosen to optimize run time and accuracy. In our implementation we have used the following values: $q = k$ and $\ell = 1$.

Algorithm 2. Deterministic Sampling-based Clustering (DSC)

Input: A set of n data points; integers p, k, and r.
Output: The best k clusters
begin

1. Divide the data points into p equal sized parts.
2. Cluster each part into q clusters.
3. Deterministically select ℓ representatives from each of the above clusters.
4. Put all of the representatives together.
5. Repeat r times steps 1 through 4.
6. Cluster the remaining points into k clusters and find the center of each of these final clusters.
7. Assign each input point x to that cluster whose center (from among all the cluster centers) is the closest to x.

5 Analysis

5.1 Time Complexity

Let there be r levels of sampling in the algorithm. Note that the standard hierarchical clustering on n points can be done in $O(n^2)$ time. Let the number of parts in level i be p_i, for $1 \leq i \leq r$. In each level and each part assume that there are q clusters and from each cluster we pick ℓ representatives.

In level 1 there are p_1 parts and a total of n points and hence each part has $\frac{n}{p_1}$ points. To cluster each part we spend $O((n/p_1)^2)$ time and hence the total time spent in level 1 is $O\left(\frac{n^2}{p_1}\right)$. From each part of level 1, we pick $q\ell$ representatives and hence the total number of points that move onto level 2 is $\frac{n}{p_1}q\ell$.

There are p_2 parts in level 2 and each part has $\frac{n}{p_1}\frac{q\ell}{p_2}$ points. As a result, the total time spent in level 2 is $O\left(\frac{n^2(q\ell)^2}{p_1^2 p_2}\right)$. Proceeding in a similar manner, the

total number of points in level j (for $1 \leq j \leq r$) is $\frac{n}{p_1 p_2 \cdots p_{j-1}}(q\ell)^{j-1}$ and there are p_j parts in this level. Therefore, the total time spent in level j is $O\left(\frac{n^2(q\ell)^{2(j-1)}}{p_1^2 p_2^2 \cdots p_{j-1}^2 p_j}\right)$.

Putting together, the total time spent in all the r levels of sampling is $O\left(\frac{n^2}{p_1} + \frac{n^2(q\ell)^2}{p_1^2 p_2} + \cdots + \frac{n^2(q\ell)^{2(r-1)}}{p_1^2 p_2^2 \cdots p_{r-1}^2 p_r}\right)$. When $q\ell$ is no more than a constant fraction of p_i for every i, then this run time simplifies to $O\left(\frac{n^2}{p_1}\right)$. Also, at level r of sampling, we identify k cluster centers and every other point is assigned to one of these clusters based on which of these centers is closest to that point. The total time for this is $O(nk)$. In summary, the total run time of the algorithm is $O\left(\frac{n^2}{p_1} + nk\right)$.

Note: In Algorithm 2 we have assumed that $p_1 = p_2 = \cdots = p_r = p$.

5.2 Accuracy

The accuracy of any (randomized or deterministic) sampling based clustering algorithm can be established by verifying that the final clusters of interest are well represented in the sample. In particular, if $C_1, C_2, \ldots, C_k$ are the clusters present in the input dataset then there should be enough representation from each of the clusters in the sample. This verification seems to be very intuitive as has been pointed out in the CURE paper [7]. We can use this observation to verify the validity of our algorithm. In the following subsections we perform this analysis in the worst case as well the average case. For simplicity we consider only one level of sampling. Specifically, we partition the input into p equal parts, cluster each part, select representatives from each cluster of each part, put together all the representatives and cluster them into k clusters, find the centers of these clusters, and assign each input point to the cluster whose center is the closest. The analysis can be extended to multiple levels as well.

Worst Case Analysis: Consider an input I consisting of n points in a high-dimensional space. Let the final clusters in I be $C_1, C_2, \ldots, C_k$. We partition I into p parts. Let these parts be $A_1, A_2, \ldots, A_p$. Clearly, the size of each part is $\frac{n}{p}$. Each A_i is clustered into q clusters and we pick ℓ representatives from each cluster. The average number of points in each such cluster is $\frac{n}{pq}$. When the number of points from some C_i is very small in some A_j, then, when A_j is clustered, we may not be able to recognize the presence of a cluster of points from C_i. Let $\tau(n)$ be the minimum number of points from C_i that should be in A_j for C_i to be detected. Let n_i^j be the number of points from C_i in part A_j, for $1 \leq i \leq k$ and $1 \leq j \leq p$. If any n_i^j is less than $\tau(n)$ then there may not be a presence of C_i in A_j, i.e., none of the cluster representatives from A_j may be from C_i. If n_i^j is $\geq \tau(n)$, then the number of representatives will be $\frac{\ell q}{(n/p)} n_i^j$. This means that the total number of representatives from C_i that are picked from all the parts $A_1, A_2, \ldots, A_p$ is at least $\frac{\ell q}{(n/p)}(n_i - p\tau(n))$, where $n_i = |C_i|$. If this number is $\geq \tau(n)$ for every i, then every cluster C_i will be recognized at the end. This happens if $n_i \geq p\tau(n) + \frac{n\tau(n)}{p\ell q}$, for every $1 \leq i \leq k$.

Average Case Analysis: Now we show that if the sample size is large enough then each final cluster will have enough representation in the sample with high probability. Please note that our algorithm employs deterministic sampling and the probability we refer to is computed in the space of all possible inputs. In other words, the verification corresponds to the average case performance of the algorithm. Before we present the verification we state the well-known Chernoff bounds.

Chernoff Bounds: If a random vraiable X is the sum of n iid Bernoulli trials with a success probability of p in each trial, the following equations give us concentration bounds of deviation of X from the expected value of np. X is said to be binomially distributed and this distribution is denoted as $B(n,p)$. By *high probability* we mean a probability that is $\geq (1-n^{-\alpha})$ where α is the probability parameter and is typically a constant ≥ 1. The first equation is more useful for large deviations whereas the other two are useful for small deviations from a large expected value.

$$Pr(X \geq m) \leq (np/m)^m e^{m-np} \tag{1}$$

$$Pr(X \leq (1-\epsilon)np) \leq exp(-\epsilon^2 np/2) \tag{2}$$

$$Pr(X \geq (1+\epsilon)np) \leq exp(-\epsilon^2 np/3) \tag{3}$$

for all $0 < \epsilon < 1$.

Accuracy Verification: Let I be a given set of n points to be clustered and let the final list of clusters be $C_1, C_2, \ldots, C_k$. Let the number of points in C_i be n_i, for $1 \leq i \leq k$. Consider a simple algorithm where we deterministically pick a sample, cluster the sample into k clusters, and assign every input point x to the cluster whose center (from among all the cluster centers) is the closest to x. Let S be a deterministic sample of size m from I. Also, let X_i be the number of points of C_i in S, for $1 \leq i \leq k$. If we assume that each input permutation is equally likely, then X_i is binomially distributed as $B\left(m, \frac{n_i}{n}\right)$. $E[X_i] = \frac{mn_i}{n}$. Using Chernoff bounds equation (2), $Pr\left[X_i \leq (1-\epsilon)\frac{mn_i}{n}\right] \leq \exp\left(\frac{-\epsilon^2 mn_i}{2n}\right)$. For $\epsilon = 1/2$, $Pr\left[X_i \leq \frac{mn_i}{2n}\right] \leq \exp\left(\frac{-mn_i}{8n}\right)$. This probability will be $\leq n^{-\alpha}$ when $m \geq \frac{8\alpha n \log_e n}{n_i}$. For this value of m, $Pr[X_i \leq 4\alpha \log_e n] \leq n^{-\alpha}$. Let $n_{min} = \min_{i=1}^{k} n_i$. In summary, when $m \geq \frac{c\alpha n \log_e n}{n_{min}}$ (for some constant $c > 8$), every C_i will have a good representation in S and hence the clusters output will be correct. For example, if $n_{min} = \sqrt{n}$, then it suffices for m to be $\geq 8\alpha\sqrt{n}\log_e n$.

6 Simulation Environment

We have evaluated the performance of DSC via rigorous simulations. Both run time and accuracy are considered. Synthetic as well as standard benchmark datasets have been employed for testing. A description of these datasets follows.

We have generated synthetic datasets based on both uniform and skewed distributions. All the datasets are from a high-dimensional space. Our datasets based on uniform distribution are generated by picking each coordinate value of each point uniformly randomly from the interval [0, 200]. To generate skew-distributed datasets, at first we generated datasets part-by-part and finally put them together. For example, a skew-distributed dataset could consist of 10 parts where each part is uniformly distributed in a different interval. The following tables [Please see Table 1 and Table 2] describe our synthetic and benchmark datasets. Benchmark datasets have been downloaded from [5].

Table 1. Synthetic Datasets

Name	Size	Attributes	Distribution
U1	11k	200	Uniform
U2	12k	200	Uniform
U3	13k	200	Uniform
U4	14k	200	Uniform
U5	15k	200	Uniform
S1	11k	200	Skewed
S2	12k	200	Skewed
S3	13k	200	Skewed
S4	14k	200	Skewed
S5	15k	200	Skewed

Table 2. Benchmark Datasets

Name	Size	Attributes	# of clusters
Thyroid	215	5	2
Wine	178	13	3
Yeast	1484	8	10
Aggregation	788	2	7
D31	3100	2	31
Flame	240	2	2
R15	600	2	15
Pathbased	300	2	3

7 Simulation Results

In this section we present our experimental results. All the programs have been run on an Intel Core i5 2.3GHz machine with 4GB of RAM.

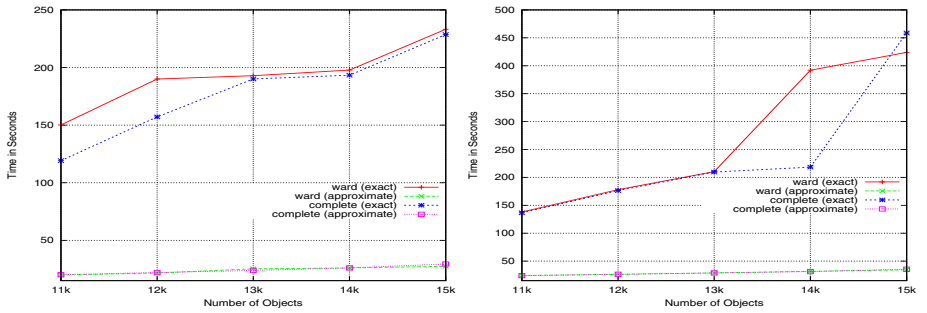
7.1 Synthetic Datasets

In our implementation of DSC we have employed one level of sampling. In particular, we partitioned the entire dataset into groups of size 500 each and clustered each group into 10 clusters. From each such cluster we picked one representative. These representatives were then put together and clustered into 10 clusters. Finally, each input point was assigned to the cluster with the closest center.

We have computed the clustering efficiencies as the average over all the synthetic datasets (uniformly distributed and skew-distributed, respectively). Also, we have applied different clustering methods [Please see Table 3] such as *ward*, *complete*, and *average*. In the case of skew-distributed datasets the average clustering efficiencies found by applying different clustering methods are in the range of [99%, 101%]. On the other hand, the efficiency exceeds 100% on uniform-distributed datasets. As the size of the datasets and/or the dimension increases, our algorithm outperforms exact algorithms, in terms of run time, by more than an order of magnitude [Please see Figure 1]. To demonstrate scalability of our

Table 3. Efficiency - Synthetic Datasets

Clustering method	Uniformly distributed datasets	Skew-distributed datasets
ward	100.42%	100.46%
complete	100.45%	100.22%
average	100.00%	99.26%



(a) Uniform data having 200 attributes (b) Skew data having 200 attributes

Fig. 1. Time to find 10 clusters from each dataset by applying various exact and approximate methods

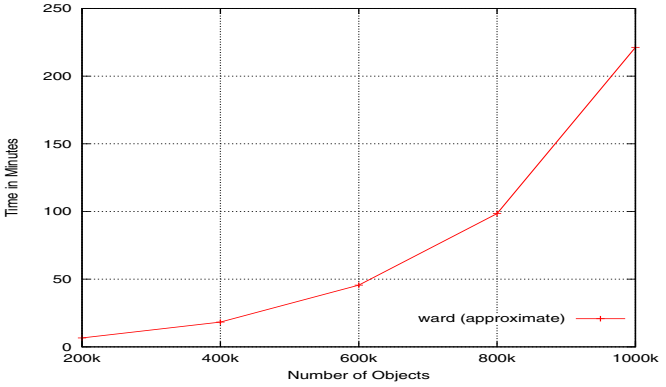


Fig. 2. Time to find 10 clusters from each of the big datasets by applying ward (approximate) method

proposed algorithm we have generated 5 big datasets occupying a large collection of objects. Each object consists of 2 attributes. We could not be able to run the exact algorithms on these big datasets using our Intel Core i5 2.3GHz machine with 4GB of RAM. On the contrary our proposed algorithm is able to perform clustering on these datasets without stalling the machine [Please see Figure 2].

We have also compared our algorithm with other well known clustering algorithms such as k-means, Partitioning Around Medoids (PAM), and Density-

Based Spatial Clustering of Applications with Noise (DBSCAN). The datasets considered here are generated by picking each coordinate value of each point uniformly randomly from the interval [0, 200]. Each point consists of 200 attributes and each of the algorithms is tuned to find 10 clusters. As the size of the datasets and/or the dimension increases, our algorithm outperforms all of the aforementioned algorithms, in terms of run time, by more than an order of magnitude and the average clustering efficiencies found by applying different clustering methods are in the range of [99%, 100%] [Please see Table 4].

Table 4. A Comparison

		k-means	**ward**		**complete**		**average**	
Objects	Dim	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)
10k	200	23.49	99.70%	18.59	99.74%	18.27	99.13%	17.93
20k		65.46	99.79%	36.04	99.76%	36.64	99.19%	36.02
50k		235.74	99.87%	94.98	99.87%	98.48	99.23%	97.16
100k		612.85	99.91%	218.96	99.91%	227.33	99.28%	237.47
		PAM	**ward**		**complete**		**average**	
Objects	Dim	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)
10k	200	121.75	99.96%	18.59	100.00%	18.27	99.39%	17.93
		DBSCAN	**ward**		**complete**		**average**	
Objects	Dim	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)	Efficiency	Time (s)
10k	200	359.62	100.00%	22.58	100.00%	22.16	100.00%	21.88

7.2 Benchmark Datasets

We have computed clustering efficiency separately for each of the benchmark datasets. For each dataset we generated the same number of clusters as shown in Table 2. A comparison has been made with various exact algorithms such as *ward*, *complete*, and *average*. From the results [Please see Table 5], we infer that the accuracy obtained by DSC is very competitive with those of exact algorithms. Also, for D31 dataset DSC is around 4 times faster than exact algorithms. When the data size is less than one thousand DSC is not faster than exact algorithms.

Table 5. Efficiency - Benchmark Datasets

Method	Thyroid	Wine	Yeast	Aggregation	D31	Flame	R15	Pathbased
ward	100.92%	98.80%	103.80%	96.89%	92.32%	97.26%	90.16%	98.71%
complete	101.72%	99.63%	116.51%	95.76%	86.79%	110.62%	81.62%	99.83%
average	101.41%	99.99%	122.12%	95.76%	94.36%	95.78%	91.21%	101.88%

8 Conclusions

Sampling is a powerful technique that has been applied to solve many fundamental problems of computing efficiently. Random sampling is much more popular

than deterministic sampling. In the context of clustering, random sampling has been successfully applied in a number of algorithms such as CURE. To the best of our knowledge, deterministic sampling has not been employed before for designing clustering algorithms. In this paper we have presented a novel deterministic sampling technique called DSC that can be used to speedup any clustering algorithm. As a case study, we have demonstrated the power of DSC in the context of hierarchical clustering. We have tested the performance of DSC on both synthetic and benchmark datasets. DSC achieves impressive accuracies. On many datasets, DSC achieves better accuracies than exact algorithms! Moreover, the speedups obtained are equally impressive. In particular, DSC is faster than exact algorithms by more than an order of magnitude. We thus feel that DSC is a very effective sampling technique. Please note that deterministic sampling is preferable over random sampling since the analysis done using deterministic sampling will always hold (instead of holding on an average or with high probability).

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. ACM-SIGMOD Conf. on the Management of Data, pp. 94–105 (1998)
2. Basu, S., Davidson, I., Wagstaff, K.: Constrained clustering: advances in algorithms. In: Theory and Applications: Data Mining and Knowledge Discovery, vol. 3. Chapman & Hall/CRC (2008)
3. Chapelle, O., Schlkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press (2006)
4. Cheung, Y.-M.: k^*-means: a new generalized k-means clustering algorithm. Pattern Recognition Letters 24, 2883–2893 (2003)
5. Clustering datasets, `http://cs.joensuu.fi/sipu/datasets/`
6. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial data sets with noise. In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, pp. 226–231 (1996)
7. Guha, S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for large data sets. In: Proc. ACM SIGMOD Conference (1998)
8. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. In: Proc. IEEE Conference on Data Engineering (1999)
9. Hinneburg, A., Keim, D.: An efficient approach to clustering in large multimedia data sets with noise. In: Proc. 4th International Conference on Knowledge Discovery and Data Mining, pp. 58–65 (1998)
10. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2, 283–304 (1998)
11. Jain, A.K., Murty, M.N., Flyn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3) (1999)
12. Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. Computer 32(8), 68–75 (1999)
13. Kashima, H., Hu, J., Ray, B., Singh, M.: K-means clustering of proportional data using L1 distance. In: Proc. Internat. Conf. on Pattern Recognition, pp. 1–4 (2008)
14. Lange, T., Law, M.H., Jain, A.K., Buhmann, J.: Learning with constrained and unlabelled data. In: IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition, vol. 1, pp. 730–737 (2005)

15. Olken, F., Rotem, D.: Random sampling from databases: a survey. Statistics and Computing 5(1), 25–42 (1995)
16. Rajasekaran, S.: Selection algorithms for parallel disk systems. Journal of Parallel and Distributed Computing 64(4), 536–544 (2001)
17. Salter-Townshend, M., Murphy, T.B., Brendan, T.: Variational Bayesian inference for the latent position cluster model for network data. Computational Statistics and Data Analysis 57(1), 661 (2013) ISSN 0167-9473
18. Sheikholeslami, C., Chatterjee, S., Zhang, A.: WaveCluster: A multi resolution clustering approach for very large spatial data set. In: Proc. 24th VLDB Conf. (1998)
19. Smet, Y.D., Nemery, P., Selvaraj, R.: An exact algorithm for the multicriteria ordered clustering problem. Omega 40(6), 861 (2012) ISSN 0305-0483
20. Tabachnick, B.G., Fidell, L.S.: Using multivariate statistics, 5th edn. Allyn and Bacon, Boston (2007)
21. Wang, W., Yang, J., Muntz, R.: STING: A statistical information grid approach to spatial data mining. In: Proc. 23rd VLDB Conference, Athens, Greece (1997)
22. Yi, X., Zhang, Y.: Equally contributory privacy-preserving k-means clustering over vertically partitioned data. Information Systems 38(1), 97 (2012) ISSN 0306-4379
23. Zhang, T., Ramakrishnan, R., Linvy, M.: BIRCH: An efficient data clustering method for very large data sets. Data Mining and Knowledge Discovery 1(2), 141–182 (1997)

Software Clustering Using Automated Feature Subset Selection

Zubair Shah[1], Rashid Naseem[2], Mehmet A. Orgun[3], Abdun Mahmood[4], and Sara Shahzad[5]

[1] Dept. of Computer Science, University of Venice, Italy
[2] Dept. of Computer Science, City University of Science and I.T., Pakistan
[3] Department of Computing, Macquarie University, Sydney, Australia
[4] University of New South Wales, Canberra, Australia
[5] Department of Computer Science, University of Peshawar, Pakistan

Abstract. This paper proposes a feature selection technique for software clustering which can be used in the architecture recovery of software systems. The recovered architecture can then be used in the subsequent phases of software maintenance, reuse and re-engineering. A number of diverse features could be extracted from the source code of software systems, however, some of the extracted features may have less information to use for calculating the entities, which result in dropping the quality of software clusters. Therefore, further research is required to select those features which have high relevancy in finding associations between entities. In this article first we propose a supervised feature selection technique for unlabeled data, and then we apply this technique for software clustering. A number of feature subset selection techniques in software architecture recovery have been proposed. However none of them focus on automated feature selection in this domain. Experimental results on three software test systems reveal that our proposed approach produces results which are closer to the decompositions prepared by human experts, as compared to those discovered by the well-known K-Means algorithm.

Keywords: Software Clustering, Feature Selection, K-Means.

1 Introduction

Software architecture provides an abstract level view of a software system which may be explained by means of different software structures [1]. Due to the software evolution, it is often the case that software systems deviate from these architectural descriptions and thus do not reflect the system's actual architecture anymore [2]. Hence, in the phases of software maintenance, resue and re-engineering, it is very difficult for software practitioners to understand a software system without having access to the actual and up-to-date architecture [3]. So it has become necessary to recover the software architecture from the available sources of information, such as source code, experts knowledge and executable files [2]. Thus many researchers have proposed and developed a number of automated software clustering techniques and tools to support the software architecture recovery [4], [5] [6]. Among all the techniques, clustering is the most dominant technique that has been used for the recovery of software architecture or modularization [7] [8] [9] [10].

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 47–58, 2013.

Clustering is a technique of making groups of similar entities (e.g., files or classes) using their features [11]. Clustering is used to make partitions into quality modules of the software systems based on the relationships/features among entities. Clustering approaches can be generally divided into two types, partition based and hierarchical [12]. Partition-based clustering makes flat clusters/partition by moving entities from one cluster to another while hierarchical clustering approaches repeatedly merge or split entities or group of entities respectively resulting in a dendrogram, a tree like structure. In this research we employ a very commonly used partition-based clustering technique, i.e., K-Means [13].

During software clustering many issues may arise [14], such as, 1) selection of features, 2) selection of similarity/distance measures, 3) selection of clustering techniques, and 4) selection of evaluation criteria. Researchers have addressed the issues 2, 3, and 4 but little work has been done on issue 1 [15]. It is significant to note that the quality of software clustering results depends on selecting suitable features of the extracted entities [15]. Increasing the number of features may increase the quality of clustering results but adding features ahead of a certain limit, may deteriorate results [1] [16]. Therefore, it is critical to select only those features which have more information regarding the entities to be clustered. To the best of our knowledge, feature subset selection for software clustering is usually done through testing of different sets of extracted features of a given software system. Such an approach has a number of shortcomings. For example, it requires domain knowledge to understand the concepts of software architecture and make useful categories of features for testing its significance in clustering. Its validity is not applicable in general since the best category of features reported for a given software may not be the best subset for another software system or it may not even exist in another software system. It is usually carried out manually by experts of the domain and testing different combinations of features requires a lot of time.

This paper proposes a feature selection technique for clustering software entities. The features extracted from the source code of a given software system are unlabeled data, which have no class label at all. Our approach first converts this data to supervised data by employing the technique of Zubair et al [17]. And then it performs automated feature selection using a well known data mining technique called Correlation-based Feature Selection Subset Evaluation (CFS) [18]. The selected subset of features is then used for software clustering by employing the K-means algorithm. To check the quality of the clusters accomplished by this method (i.e., using reduced features), we have compared it with those accomplished using the full set of features. The results revealed that the quality of clusters is better when only a subset of features is used. The approach does not require domain knowledge, it is faster and it would be valid for all types of software systems.

The rest of the paper is organized as follows: Section 2 presents the work related to software clustering. An overview of our proposed approach is given in Section 3, together with discussion of supervised feature selection method for software clustering. In Section 4, we present our experimental design and setup that we have adopted to test our proposed approach. In Section 5, the experimental results of our proposed approach on three software systems are presented, followed by a discussion of threats to the validity of our study. To end, we conclude and discuss future work in Section 6.

2 Related Work

Architecture recovery requires extraction of different types of features followed by grouping (clustering) software entities into meaningful structures based on those features. In this section we discuss those approaches that are most closely related to our work.

To retrieve the software architecture (module view) Bittencourt and Guerrrero [19], presented an empirical study of K-Means, Edge betweenness, design structure matrix, and modularization quality clustering algorithms. They evaluated these algorithms using the Authoritativeness, extremity of clusters and stability, which were quantified against using four software systems. They concluded that KM outperforms in terms of authoritativeness and extremity. In a similar study [20], the authors implemented clustering approaches including K-Means, Edge betweenness, modularization quality and design structure matrix clustering as a plug-in for Eclipse. Plug-in was applied on three different open source software systems. The results were assessed and concluded that K-Means produces better results in terms of authoritativeness as compared to other existing algorithms in the literature. Similarly, Corazza et al. [21], proposed partition based clustering approach based on lexical information extracted from Java classes. These information/features were weighted using a probabilistic approach; Expectation-Maximization (EM) algorithm is applied. K-Medoids clustering algorithm has been implemented to make groups of the classes. The approach is evaluated using authoritativeness and extremity of clusters. The results revealed that K-Medoids improve the results while using EM for feature weighting. This work has been extended in [22], by investigating the six different types of lexical features introduced by developers, namely; statements of source code, comments, and class, data, method and actual parameter names. An empirical study has been conducted on 13 open source software systems.

In contrast to our work, all of the above techniques need human intervention in one way or another during the clustering process, while we just need labeling of the entities for feature selection. The technique introduced by Corazza et al. [21] [22], needs comments about source code of the software systems. In our automated feature selection approach we do not rely on the type of features, since we directly apply a feature selection technique to reduce or select highly related subsets of the features for a given software system.

Basic clustering algorithms like Single Linkage (SL) and Complete Linkage (CL) have also been used for software clustering. For example, Wiggerts [23], proposed clustering algorithms, like CL and SL. He suggested a careful selection of the entities, type or number of features and similarity or distance measures for software clustering. The similar work by Anquital and Lethbridge has been extended in [14]. They presented a detailed comparative analysis of different hierarchical clustering algorithms, for example CL, SL, Weighted Average Linkage and Unweighted Average Linkage algorithms. To evaluate these algorithms precision, recall, cohesion, coupling and size of the clusters were used. They also analyzed types of features and similarity measures. They concluded that CL linkage is a better choice for remodularization. Indirect features may produce good quality of clusters as compared to direct features, when employing basic clustering algorithms.

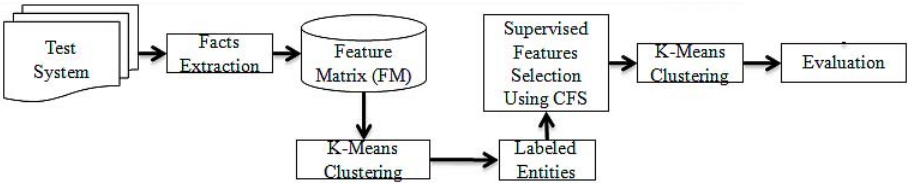


Fig. 1. System Diagram for Software Clustering Approach

Saeed and Maqbool [8], proposed a new algorithm known as "Combined algorithm". For newly formed clusters, the Combined algorithm makes a new feature vector by taking binary OR. The proposed algorithm is compared with existing clustering algorithms using precision and recall. Experiments conducted on five subsystems of Xfig, showed that the Combined algorithm performed better. In a followup work [24], the authors proposed "Weighted Combined" (WC), a new hierarchical clustering algorithm, which overcomes the deficiencies in the Combined algorithm. It makes a new feature vector by maintaining the number of entities accessing a feature in a cluster. WC produced better results as compared to the Combined algorithm, using precision, recall and the number of clusters as assessment criteria.

Naseem et al. [25] and [9], proposed two new similarity measures for software clustering, i.e., Jaccard-NM and Unbiased Ellenberg-NM respectively. They used proprietary industrial software systems for experimental purposes. Applying these measures with the CL algorithm, produced better results. In [26], cooperative approaches were proposed, where two measures cooperate with each other in a hierarchical clustering algorithm. These approaches generate better clustering results as compared to a clustering approach with a single measure. They used MoJoFM and arbitrary decisions as evaluation criteria.

3 Our Approach to Software Clustering

The system diagram for our software clustering approach is given in Fig.1. The process of software clustering starts with the extraction of the entities and the relationships between them as features from the source code of the given system. This is followed by K-Means clustering for assigning labels to the extracted entities. Those labeled features are fed to a supervised feature selection technique using Correlation-based Feature Selection Subset Evaluation (CFS) and the selected features are in turn used in K-Means clustering resulting in software clusters which are then evaluated against expert decomposition using evaluation criteria.

3.1 Clustering

Clustering is an unsupervised method that is widely used for different application areas. The most important intention of clustering is to identify the fundamental unknown structures in data for many reasons like discovery of unknown groups and recognizing patterns. It employs the process of, grouping a set of entities into clusters of similar

entities [5]. Thus, a cluster is a group of entities that are similar to each other and are different to the entities in other clusters. Similarity or distance can be calculated for pairs of entities based on a set of features. In the context of our experiment, similarity is a measurement of a degree which reflects the strength of the association between two entities, and distance measures the difference between two entities. We have used Euclidean distance measure as shown in Eq.1, as the distance measure. It calculates the root of the square differences between all the features **f** of the pair of entities:

$$distance(X, Y) = \sqrt{b + c} \tag{1}$$

where, a, b, c and d can be found using Table 1. Assume that we have two entities X and Y, then a is the count of features that are present in both entities, b and c show the number of features that are present in one entity and absent in the other and d is the number of features that are absent in both entities. The total number of features can be represented as n.

Table 1. Contingency table

		Y		
		1	0	Sum
X	1	a	b	a+b
	0	c	d	$c + d$
	Sum	$a + c$	$b + d$	n

Table 2. An example feature/(E x f) matrix

	fr1	fr2	fr3	fr4	fr5	fr6
E1	1	0	0	1	1	0
E2	1	0	1	1	0	0
E3	1	1	0	0	0	0
E4	1	1	1	1	1	0

In the context of software clustering, features are generally in binary form, i.e., they show the occurrence (presence) or lack (absent) of a feature [15]. An example feature matrix containing 4 entities and 6 features is presented in Table 2. All features are indirect. Each feature may be shared or not shared between the entities. If a feature is shared then it will be marked as present, i.e., 1, otherwise, absent, i.e., 0. As can be seen in Table 2, fr1 is absent in entity E1 while present in all the other entities. It may also be the case that a feature may exist in a software system but it may not used by any entity, for example, no entity is using fr6, therefore it is marked as absent for all the entities.

3.2 Feature Selection Technique

There are basically two steps in selecting the best subset of features from the types of data sets discussed above.

Step 1 (Producing Labeled Data). In this step we use the technique described in [17] which applies the K-Means clustering algorithm to obtain the initial labels of the data. K-Means clustering algorithm produces cluster labels such as *cluster0, cluster1, cluster2* etc of each entity depending on the value of *K* (*K* is the number of clusters in K-Means). The output produced by K-Means is our labeled data with class labels as *cluster0, cluster1, cluster2* etc. The labels produced by K-Means are not accurate as K-Means' accuracy may not be 100 % but these labels could be used as an estimation

to correct labels. Our approach assumes these estimated labels produced by K-Means as correct labels of the data and in this way we are able to convert unsupervised data to supervised data.

Step 2 (Correlation-based Feature Selection Subset Evaluation (CFS)). It assesses the significance of a subset of features by allowing for the individual predictive capability of every feature along with the measurement of redundancy among each feature [18]. It chooses a reduced set of features that are highly connected with the expected class information (for more information see [18]). We used two types of search methods in conjunction with CFS described below:

- Best First (BF) uses greedy hillclimbing improved with backtracking ability to search for a subset of features. The arrangement of successive non-improving nodes permitted control the level of backtracking completed. BF starts with the vacant set of features and looks for features, or begins with the full set of features and looks for features toward the back, or initiates at any position and looks for in equal directions.
- Greedy Step Wise (GS) carries out a greedy forward or backward search through the space of features subsets. This approach may start with no features or all features or from any feature. It stops when the addition or deletion of any leftover features results in a quality decrease.

3.3 K-Means Algorithm

The K-Means (KM) algorithm is a very well known and widely employed clustering technique. K in the name represents the number of clusters to be determined before KM starts. Each cluster has a mean of its entities, the so called centroid. To assign each entity to its nearest centroid, the distance or similarity is to be computed between the entity and its centroid. A very commonly used Euclidean distance measure can be used to compute the distance between an entity and a centroid. After finding the distances for each entity with the centroids, entities are assigned to their nearest centroid. The centroid is recalculated using the mean for the newly made cluster and the whole process is iterated until a stopping criterion is achieved.

4 Experimental Design

This section describes the experimental procedure we adopted in our study. The procedure starts with data set selection, feature extraction and selection, software clustering and finally evaluation.

4.1 Data Sets

We selected two proprietary and one open source object oriented software systems to conduct our empirical study. We have also tested our aproach on other software systems, but due to the space and time limitations, it is not possible to include them in this article. Those industrial software systems are:

1. **Statistical Analysis Visualization Tool (SAVT)** is a software system which offers utilities associated to arithmetical data and visualization of the results. It has been developed using Visual C++ language. The system has 27,311 lines of source code and comprises of 97 classes.
2. **Fact Extractor System (FES)** is a fact extraction tool that parses software systems developed in Visual C++ and extracts information regarding entities, features and other statistics about the systems. This software is developed in Visual C++ and has 10402 lines of source code with 47 classes.
3. **Mozilla** is a open source software system for Internet browsing. We used Mozilla version 1.3 released in March 2003. Similar to the approach taken in [27], we preferred 6 subsystems over 10 subsystems of Mozilla which have 258 Files, similar to the approach adopted in [27].

4.2 Feature Extraction and Selection

Fact Extractor System of [28] was used to extract detailed design information from source code through static analysis. The information extracted was, entities and relationships from the source code of the systems in Visual C++. Detailed design extracted for Mozilla is taken from [29] and the details of the fact extraction method are given in [5]. They extracted files as entities and the relationships among files as features. The number of features are shown in Table 4, taken from [27] [15].

Having extracted the features from source code, we have adopted two approaches for software clustering using 1) full set of features and 2) reduced set of features using feature selection techniques, as given in Table 3. In the first approach we used the full set using the KM algorithm, as given on serial number 1 in Table 3. In the second approach we used a feature selection technique (described in Section 3) before applying the KM algorithm, as shown on serial numbers 2 and 3.

4.3 Clustering Algorithm

The next phase of our experimental setting is to apply a clustering algorithm. Before applying clustering algorithms, distance/similarity measure to be calculated between the entities to find the association among entities. We calculated the dissimilarity between pairs of entities using the Euclidian distance measure. Then we used the K-Means algorithm for software clustering.

Table 3. Clustering strategies

Sr. No	Strategy	Algorithm	Feature Selection Method	Weka Name	Search Method
1	KM	KM	Nil	Nil	Nil
2	KCB	KM(K)	Correlation-based Feature Selection Subset Evaluation (C)	CfsSubsetEval	Best First (B)
3	KCG	KM(K)	Correlation-based Feature Selection Subset Evaluation (C)	CfsSubsetEval	Greeedy Step Wise (G)

Table 4. Statistics of Relationships among Entities

Feature Types	Count		
	Mozilla	SAVT	FES
Inheritance	-	986	166
Containment	-	1032	56
Class in Methods	-	1900	384
Same Generic Class	-	49	91
Same Generic Parameter	-	0	4
File	-	264	42
Total	258	4231	743

4.4 Assessment

To assess the results, external evaluation has commonly been used in the literature. In external evaluation, the automatically obtained result is compared with already prepared decomposition by a human expert. One of the most common ways to evaluate the results is MoJoFM [30]. This measure calculates the percentage of Moves and Joins required, converting automatically obtained results into expert decomposition. If we have result R and expert decomposition D, then MoJoFM is given by:

$$MoJoFM = \left(1 - \frac{mno(R,D)}{max(\forall mno(R,D))}\right) * 100 \tag{2}$$

where mno(R, D) is the least number of 'move' and 'join' operations required to change from R to D and max(∀ mno(R, D)) is the maximum of the minimum number of likely 'move' and 'join' operations required to change from R to D. MoJoFM values lie between 0% and 100%. A higher MoJoFM value indicates higher similarity between the automated and human prepared decompositions and therefore good quality results, whereas a lower MoJoFM value indicates a lower similarity.

Second, Precision and Recall are also used widely to compare the automated result with expert decomposition. This method has been used to check the effectiveness of retrieved results [24]. Precision is the fraction of pairs of entities in the automatically obtained result that also exist in the decomposition prepared by the human expert, where pair of entities, must be in the same cluster. If X is the set of intra pairs of the automated results and Y is the set of intra pairs of the expert decomposition then precision is given by:

$$Precision = \frac{|X \cap Y|}{X} \tag{3}$$

Recall can be defined as the percentage of the pair of entities in the expert decomposition which are also in the automatically obtained result.

$$Recall = \frac{|X \cap Y|}{Y} \tag{4}$$

It is very desirable that Recall is equal to 100% but this is not the case because there is a tradeoff between Precision and Recall [24]. Higher Precision and Recall values

will produce better results. To overcome this phenomena of a tradeoff, we also used F-measure which is the harmonic mean of Precision and Recall, which can be calculated as given in equation 5. Higher the value of F-measure, the better the results will be.

$$F\_Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

5 Experimental Evaluation

This section presents the empirical results of the KM algorithm and **F**eature **S**election method with **KM** (FSKM, i.e., KCB and KCG) for the three data sets used in our experiments. In order to assess the quality of the recovered architectures for the used data sets, we compared them against expert decompositions taken from [27][15].

5.1 Experimental Results

The experimental results of KM and FSKM using MoJoFM are shown in the Table 5 for the three data sets and four techniques. MoJoFM is the most significant of the four evaluation measures used in this paper. This measure reflects the best value of clustering results as compared to other measures for software clustering.

This section also contains some results of the Complete Linkage (CL) algorithm when they were available. CL is a hierarchical clustering algorithm, used for software clustering. For Mozilla using CL, the values are taken from [26], for other datasets using CL, the values are taken from [25]. The values of CL are shown in the fifth column of the Table 5. All these values are recorded at the highest value of MoJoFM in the iterations.

Table 5. MoJoFM values

	KM	KCB	KCG	CL
Mozilla	34	48	48	63
SAVT	54	58	58	53
FES	47	55	55	36
Average	49	56	56	49

Table 6. Precision values

	KM	KCB	KCG
Mozilla	0.21	0.4	0.4
SAVT	0.39	0.5	0.5
FES	0.23	0.29	0.29
Average	0.41	0.5	0.5

As can be seen from Table 5, FSKM performs better as compared to KM on Mozilla, SAVT and FES. CL also generates better results for Mozilla only while its results deteriorated on other data sets as compared to other techniques. On average FSKM, i.e., KCB and KCG perform better.

The results of KM and FSKM using Precision are given in Table 6. It can be concluded from the results that the precision values for KCB and KCG are better than the other approaches used. Also, on average KCB and KCG have higher values.

Table 7 shows the values of Recall for all the techniques. It can be seen that for Mozilla all the techniques have created equal results while for SAVT, FSKM produces better results. This is the only measure for which KM has higher value for FES.

The values for F-measure are given in Table 8. F-measure takes the harmonic mean of Precision and Recall, therefore normalized the values. Now, as can be seen that for all data sets FSKM produces better results. On average FSKM (KCB and KCG) outperform KM.

Table 7. Recall values

	KM	KCB	KCG
Mozilla	0.54	0.54	0.54
SAVT	0.54	0.57	0.57
FES	0.4	0.32	0.32
Average	0.51	0.5	0.5

Table 8. F-Measure values

	KM	KCB	KCG
Mozilla	0.31	0.46	0.46
SAVT	0.45	0.53	0.53
FES	0.29	0.3	0.3
Average	0.43	0.49	0.49

5.2 Discussion

For our experiments we have used three software systems, which were also used in [25] [9] [15] [27]. It would be more instructive to perform extensive experiments on other test software systems to generalize our opinions according to our proposed feature selection criteria. Furthermore, to generalize our results, it is required to do experiments with test software systems in a range of size, domains and development languages (e.g. Java, C++ and C#). Our three test software systems were two different industrial proprietary systems and an open source system. However, we focused on the test software systems which were medium in size. This is logical because in real projects throughout the maintenance phase, developers carry out their job on a small part of the entire software system [31]. For fair and clear experiments, we used library of Weka [32] for software clustering and evaluation.

There also exist some internal assessment criteria, for example stability and the number of clusters. We did not empirically test the techniques for these criteria, because we have used only the KM algorithm for software clustering. Moreover, the number of clusters is already known when employing KM.

To evaluate methods, software decompositions are necessary to be developed by a human expert, which may introduce subjectivity. Therefore we have taken all of the expert decompositions from already published worked so that relative performance of different methods could be evaluated using the same benchmarks.

6 Conclusion

This work assessed and compared two types of approaches in the perspective of software clustering. One approach uses KM on the full set of features and the second approach provides labels to unlabeled data using KM, and then uses supervised feature selection technique implemented in Weka and then applies KM on this reduced set of features. We used three software systems as data sets for evaluation, two proprietary and one open source software system, i.e., Mozilla.

Evaluation aspect was external assessment only, in which we consider MoJoFM, Precision, Recall and F-Measure. From our empirical analysis, we concluded that

using supervised feature selection approaches can improve the clustering results. Using MoJoFM, Recall and F-Measure, our approach produces significantly better clustering quality, in terms of the closeness of automated results to the one prepared by human experts, as compared to the simple KM and Complete Linkage algorithms.

Future work will investigate other feature selection methods and other types of clustering algorithms for software clustering. Work is also required to test the approaches empirically on other software systems.

References

1. Maqbool, O., Babri, H.A.: Hierarchical clustering for software architecture recovery. IEEE Transactions on Software Engineering 33(11), 759–780 (2007)
2. Wang, Y., Liu, P., Guo, H., Li, H., Chen, X.: Improved hierarchical clustering algorithm for software architecture recovery. In: International Conference on Intelligent Computing and Cognitive Informatics, pp. 247–250 (2010)
3. Fontana, F.A., Zanoni, M.: A tool for design pattern detection and software architecture reconstruction. Information Sciences 181(7), 1306–1324 (2011)
4. Mitchell, B.S., Mancoridis, S.: On the automatic modularization of software systems using the BUNCH tool. IEEE Transactions on Software Engineering 32(3), 193–208 (2006)
5. Andritsos, P., Tzerpos, V.: Information theoretic software clustering. IEEE Transactions on Software Engineering 31(2), 150–165 (2005)
6. Cui, J., Chae, H.: Applying agglomerative hierarchical clustering algorithms to component identification for legacy systems. Information and Software Technology 53(6), 601–614 (2011)
7. Mahdavi, K., Harman, M., Hierons, R.: A multiple hill climbing approach to software module clustering. In: Proceedings of the International Conference on Software Maintenance, pp. 315–324 (2003)
8. Saeed, M., Maqbool, O., Babri, H.A., Hassan, S., Sarwar, S.: Software clustering techniques and the use of combined algorithm. In: Proceedings of the European Conference on Software Maintenance and Reengineering, pp. 301–306 (2003)
9. Naseem, R., Maqbool, O., Muhammad, S.: An improved similarity measure for binary features in software clustering. In: Proceedings of the International Conference on Computational Intelligence, Modelling and Simulation (CIMSim), pp. 111–116 (September 2010)
10. Shtern, M., Tzerpos, V.: Clustering methodologies for software engineering. In: Advances in Software Engineering 2012 (2012)
11. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Survey 31(3), 264–323 (1999)
12. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2006)
13. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Applied Statistics 28 (1979)
14. Anquetil, N., Lethbridge, T.: Experiments with clustering as a software remodularization method. In: Proceedings of Sixth Working Conference on Reverse Engineering, pp. 235–255 (1999)
15. Siraj, M., Maqbool, O., Abbasi, A.: Evaluating relationship categories for clustering object-oriented software systems. IET Software 6(1), 260–274 (2012)
16. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley (2000)
17. Shah, Z., Mahmood, A.N., Mustafa, A.K.: A Hybrid approach to improving clustering accuracy using SVM. In: 2013 8th IEEE Conference on Industrial Electronics and Applications (ICIEA) (2013)

18. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. PhD thesis, The University of Waikato (1999)
19. Risi, M., Scanniello, G., Tortora, G.: Architecture recovery using latent semantic indexing and k-means: An empirical evaluation. In: 2010 8th IEEE International Conference on Software Engineering and Formal Methods (SEFM), pp. 103–112 (2010)
20. Scanniello, G., Risi, M., Tortora, G.: Architecture recovery using latent semantic indexing and k-means: an empirical evaluation. In: 2010 8th IEEE International Conference on Software Engineering and Formal Methods (SEFM), pp. 103–112. IEEE (2010)
21. Corazza, A., Di Martino, S., Scanniello, G.: A probabilistic based approach towards software system clustering. In: 2010 14th European Conference on Software Maintenance and Reengineering (CSMR), pp. 88–96 (2010)
22. Corazza, A., Martino, S., Maggio, V., Scanniello, G.: Investigating the use of lexical information for software system clustering. In: 2011 15th European Conference on Software Maintenance and Reengineering (CSMR), pp. 35–44 (2011)
23. Wiggerts, A.: Using clustering algorithms in legacy systems remodularization. In: Proceedings of the 4th Working Conference on Reverse Engineering, pp. 33–43 (1997)
24. Maqbool, O., Babri, H.A.: The weighted combined algorithm: a linkage algorithm for software clustering. In: Proceedings of the European Conference on Software Maintenance and Reengineering, pp. 15–24 (2004)
25. Naseem, R., Maqbool, O., Muhammad, S.: Improved similarity measures for software clustering. In: Proceedings of the European Conference on Software Maintenance and Reengineering, pp. 45–54 (March 2011)
26. Naseem, R., Maqbool, O., Muhammad, S.: Cooperative clustering for software modularization. Journal of Systems and Software 20 (in press, 2013)
27. Siddique, F., Maqbool, O.: Analyzing term weighting schemes for labeling software clusters. IET Software 6(3), 260–274 (2012)
28. Abbasi, A.Q.: Application of appropriate machine learning techniques for automatic modularization of software systems. Mphil. thesis, Quaid–i–Azam University Islamabad (2008)
29. Andreopoulos, B., An, A., Tzerpos, V., Wang, X.: Clustering large software systems at multiple layers. Information and Software Technology 49(3), 244–254 (2007)
30. Wen, Z., Tzerpos, V.: An effectiveness measure for software clustering algorithms. In: Proceedings of 12th IEEE International Workshop on Program Comprehension, pp. 194–203 (2004)
31. Abbes, M., Khomh, F., Guéhéneuc, Y., Antoniol, G.: An empirical study of the impact of two antipatterns, blob and spaghetti code, on program comprehension. In: 2011 15th European Conference on Software Maintenance and Reengineering (CSMR), pp. 181–190 (2011)
32. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)

The Use of Transfer Algorithm for Clustering Categorical Data

Zhengrong Xiang and Lichuan Ji

College of Computer Science, Zhejiang University
zolaxiang@gmail.com

Abstract. We propose a new method for clustering categorical data. Clustering algorithms need to be designed specifically for categorical data because it has a different nature from numerical data. Here our focus is on the partition paradigm of algorithms. One existing approach is to transform categorical data into binary data and then use k-means. However it's computationally inefficient. Another approach is k-modes, which extends k-means by replacing means with modes. In our work, we show that the center-based objective function of k-modes can not produce accurate clustering results. Instead, we propose an objective function that is generalized from the k-means objective, but not based on centers. We show that it's more effective than the center-based objective and demonstrate it with real-life datasets. We also find that by using a particular algorithm called transfer algorithm, the proposed objective function can be efficiently solved. Thus our method is both efficient and effective.

Keywords: clustering, categorical data, transfer algorithm.

1 Introduction

Clustering is a major topic in unsupervised learning. The goal is to find structure in data by grouping similar objects together. A good review on clustering is [1]. Categorical data is different from numerical data in that the feature values are nominal, e.g. *color* is an attribute of *flower* that has values of *red*, *yellow*, *purple*, etc.. Categorical values don't have explicit dissimilarity measures, e.g. the dissimilarity between *red* and *yellow* is not available. Distance measures between categorical objects are usually based on co-occurrence of attribute values [9]. This requires us to design algorithms specifically for categorical data.

Several non-partition algorithms for clustering categorical data are [2–6]. In this paper, we focus on the partition paradigm, in which k-means is the most popular one [7]. In k-means, the distance measure is Euclidean distance, thus it does not apply for categorical data. A straightforward approach [8] is to convert categorical attributes into binary attributes—creating one binary attribute for each categorical value. For example, an object of (color: red, shape: round) is converted to (red: 1, yellow: 0, green: 0, round: 1, ellipse: 0). Then k-means can be used on the binary attributes. Obviously, the number of features grows much

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 59–70, 2013.

larger, especially when the numbers of categories are large, making it a slow algorithm. Another drawback of this method is about the dissimilarity measure. It is restricted to use Euclidean distance on the transformed binary data, which is equivalent to the simple matching distance on categorical data. In practice, other dissimilarity measures might be more suitable for specific datasets, as surveyed in [9].

A closely related algorithm, k-medoids [10], can also be used on categorical data. It has the same procedure as k-means, except that the center is medoid. A medoid is the object whose average dissimilarity to all the objects in the cluster is minimal. Because the computation of medoids costs too much time, it's rarely used in practice.

Several k-means-like algorithms were proposed specifically for categorical data. They define a center, a dissimilarity measure between centers and objects, and the optimization method goes exactly like k-means. In *k-modes* [11], the center is called mode, which has the same form as an object. Each attribute of the mode takes the value whose frequency is the highest among all values. Objects are assigned to nearest modes according to the dissimilarity measure. Thus the objective function is $F = \sum_{k=1}^{K} \sum_{i \in C_k} dissimilarity(x_i, M_k)$, where x are objects and M are modes. Let's see the following example.

Example 1: The objects have three attributes: A1, A2, and A3. Table 1 shows two clusters, each has five objects. In the first cluster, the attribute value 'a' appears three times in attribute A1, more than 'b' which appears two times. Thus the mode takes 'a' as the value for attribute A1. Similarly, the mode takes 'p' and 'x' for attribute A2 and A3. In the second cluster, the mode is also [a,p,x].

Table 1. Two different clusters with a same mode: M=[a,p,x]

	A1	A2	A3
1	a	p	x
2	a	p	y
3	a	p	z
4	b	q	w
5	b	q	x

	A1	A2	A3
1	a	p	x
2	a	p	y
3	a	p	z
4	b	q	w
5	c	r	x

In the work of [12, 17], k-modes is extended by adding new information into modes. For each attribute value of a mode, the frequency of the value is also saved and then used in computing distances between objects and modes. In Example 1, the modes are both [a: 3/5, p: 3/5, x: 2/5].

Unlike the binary method, defining centers specifically for categorical data makes the partition algorithm efficient. However, we argue that a mode *under-represents* a cluster, thus the objective function is not quite *informative*. A mode has only one attribute value for each attribute to represent a cluster, while information of other attribute values is neglected. The consequence is: in the

clustering process, distances between objects and modes can not represent the true distances between objects and clusters well. For example, assume *red* is the most frequent value for the color attribute, then the information of other colors is neglected. It does not matter whether there are more *yellow* than *blue*, or how much more. See the following example.

Example 2: Continue with Example 1. We will use a popular dissimilarity measure called simple matching (also called overlap). For two objects x and y, the simple matching dissimilarity is:

$$d(x,y) = \sum_{j=1}^{A} d_j(x_j, y_j) \quad d_j(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Where A is the number of attributes. In the first cluster $C1$, $d(x_1, M) = 0$, $d(x_2, M) = d(x_3, M) = 1$, $d(x_4, M) = 3$, $d(x_5, M) = 2$. Thus objective function $F_{C1} = \sum d(x_i, M) = 7$. Then we can see $F_{C1} = F_{C2}$. It means that in the clustering process, the two clusters are treated as having the same quality of coherence. This is not true if we manually analyze the clustering quality: the only difference between the two clusters is object 5. In cluster C1, object 5 is quite similar with object 4, while cluster C2 doesn't have this similarity. It means that cluster C1 actually has a better clustering quality, which is not detected by the uninformative objective function of k-modes.

Another algorithm that is originated from k-means is called *k-representatives* [13]. The center, named as representative, is a list of all the attribute values in the cluster and their frequencies. For example, of the first cluster in Example 1, the representative is $[a : 3/5,\ b : 2/5, p : 3/5, q : 2/5, x : 2/5, y : 1/5, z : 1/5, w : 1/5]$. This definition of center is equivalent to the mean in the binary method, but the dissimilarity measure is different from Euclidean distance. The similarity (dissimilarity is the complement of similarity) between an object and a representative is to sum over frequencies of all attribute values that the object takes.

A representative has all the information of a cluster, and empirical experiments show that the clustering results are more accurate than k-modes on some benchmark datasets. However, this algorithm does not always converge. Bai [14] formally proved this, and the experiment shows that the values of objective function does not always decrease in the clustering process. Informally, convergence can be proved if for a specific distance measure, there is a center that minimizes the total distance of a cluster [7]. In fact, k-means-like algorithms can be developed for only four metric spaces: $L1$, $L2$, $L\infty$, and $L0$ (the standard Minkowski power metric where $p \to \infty$). It stems from the simple fact that these four metric spaces have calculable cluster centers (the median, mean, midrange, and mode, respectively). In k-representatives, since the center is equivalent to the mean in the binary method, the distance measure that minimizes the total distance should be Euclidean distance, not the defined distance or any other measures.

In our work, the objective function is generalized from the k-means but no longer center-based (Section 2). Thus our method avoids the uninformative weakness of k-modes, producing better clustering results. Also, we show that due to

the nature of categorical data, the transfer algorithm is efficient for the proposed objective function (Section 3). Thus it's computationally faster than the binary method. Finally, the transfer algorithm always converges to local optima. In Section 4, experiments show the superior effectiveness and efficiency.

2 The Proposed Objective Function

In this section, we first show that for numerical data, the objective function of k-means is equivalent to the so-called within-cluster dispersion. Then we generalize it for categorical data to have our proposed objective function, and show its superiority by example.

In k-means of the numerical data, If $\mathbf{X}_{N\times P} = \{x_{ij}\}_{N\times P}$ denotes the $N \times P$ data matrix (N objects on P variables), n_k is the number of objects in cluster k, the objective function we attempt to minimize is the error sum of squares (SSE, [7]).

$$SSE = \sum_{j=1}^{P}\sum_{k=1}^{K}\sum_{i\in C_k}\left(x_{ij} - \overline{x}_j^{(k)}\right)^2 \quad Where \quad \overline{x}_j^{(k)} = \frac{1}{n_k}\sum_{i\in C_k} x_{ij} \tag{2}$$

Define $\mathbf{W}_k$ as the within-cluster sums-of-squares-and-cross-products matrix for the cluster k [7]:

$$\mathbf{W}_k = \frac{1}{2n_k}\sum_{i\in C_k}\sum_{i^*\in C_k}(x_i - x_{i^*})(x_i - x_{i^*})' \tag{3}$$

As shown in [7], the SSE objective is equivalent to the sum of the traces of $\mathbf{W}_k$:

$$SSE = \sum_{k=1}^{K} tr(\mathbf{W}_k) \tag{4}$$

It's trivial to see that the trace of $\mathbf{W}_k$ (Notated as W_k) is

$$W_k = tr(\mathbf{W}_k) = \frac{1}{2n_k}\sum_{i\in C_k}\sum_{i^*\in C_k}(x_i - x_{i^*})^2 \tag{5}$$

W_k is the total distances of all pairs of objects in cluster k, divided by the number of objects in cluster k. We call it within-cluster dispersion of cluster k. Thus we have:

$$SSE = W = \sum_{k=1}^{K} W_k \tag{6}$$

Where W is the within-cluster dispersion of a whole partition.

The classic k-means procedure is to iteratively compute centers and relocate objects to the nearest centers. The equivalence of (6) tells us that when we use

this procedure, we are also minimizing the within-cluster dispersion W. In k-modes, however, the definition of center and dissimilarity measure are different from k-means, thus a similar equivalence does not hold: the within-cluster dispersion under the new dissimilarity measure is not the same as the center-based objective. As shown in Section 1, the center-based objective of k-modes is not very informative, so we can try the within-cluster dispersion instead.

By replacing the Euclidean distance in (5) with a general distance measure $d(x, y)$ for categorical data, we generalize the within-cluster dispersion as our proposed objective function:

$$W = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{i \in C_k} \sum_{i^* \in C_k} d(x_i, x_{i^*}) \tag{7}$$

A general distance measure is a desired quality in practice. While for numerical data we usually use Euclidean distance, it's not clear which dissimilarity measure is universally good for all categorical datasets [9]. To achieve good clustering results, it's better to use different measures for specific datasets.

To illustrate the advantage of this objective function comparing with the center-based objective of k-modes, we have the following example.

Example 3: Continue with Example 1. Again for the simple matching dissimilarity (1), the within-cluster dispersion of the two clusters are: $W_{C1} = 42/10$, $W_{C1} = 46/10$. It indicates that cluster C1 has better clustering quality, which is intuively correct and also our desired result. The mechanism is that the objective function (7) captures the dissimilarity between object 4 and 5. As shown in Example 2, k-modes can not capture this *subtle* information.

Now we have proposed our objective function (7) and shown its effectiveness. In Section 3, we present how to efficiently solve it using the transfer algorithm.

3 Transfer Algorithm for Clustering Categorical Data

Transfer algorithm [15, 16] is a general class of partition clustering algorithm. The structure is as follows. When an initial partition is given, the algorithm scans the dataset, and transfers(relocates) objects to other clusters if it improves a particular objective function. The process terminates when no more transfers can improve the objective.

Like k-means, transfer algorithm gives only local optima. The differences are in two ways: First, transfer algorithm does not compute cluster means or centers in the optimization process. As in the discussion of k-modes, the deficit of center-based objectives is avoided. Second, the objective function of transfer algorithm can be any function that is sound, including the SSE of k-means. As we have argued, this is a good quality for categorical data: in practice we should use different dissimilarity measures for specific datasets. And the different objective functions can all be solved by transfer algorithm.

From Tarsitano [16], there are three types of transfers. One is to transfer one object at a time to a new cluster. Another is to swap two objects into each other's

original cluster. Swapping can be used independently. It can also be combined with the first method: when the first type of transfer comes to an end that no more transfers are eligible, there are possible swaps that can further improve the objective. The third type is to reassign several objects simultaneously, which is less common. In this paper, we use only the first type of transfer to illustrate the merits of transfer algorithm for categorical data.

Given an initial partition, we try to improve the objective by relocating objects to new clusters. An object t should be relocated from cluster P to another cluster Q, only if the relocation reduces the within-cluster dispersion objective (7), that is,

$$W_{Q+t} + W_{P-t} < W_P + W_Q \tag{8}$$

Rearranging (8):

$$\Delta W = (W_{Q+t} - W_Q) + (W_{P-t} - W_P) < 0 \tag{9}$$

We call (9) the **transfer test**.

Use the definition of W, we get:

$$\begin{aligned} W_{Q+t} &= \frac{1}{2(n_Q+1)} \sum_{x_i \in Q+t} \sum_{x_{i'} \in Q+t} d(x_i, x_{i'}) \\ &= \frac{1}{2(n_Q+1)} \left(\sum_{x_i \in Q} \sum_{x_{i'} \in Q} d(x_i, x_{i'}) + \sum_{x_i \in Q} d(t, x_i) \right) \\ &= \frac{1}{2(n_Q+1)} \left(2n_Q W_Q + \sum_{x_i \in Q} d(t, x_i) \right) \end{aligned} \tag{10}$$

Similarly,

$$W_{P-t} = \frac{1}{2(n_P-1)} \left(2n_P W_P - \sum_{x_i \in P} d(t, x_i) \right) \tag{11}$$

After some simple arithmetic, the transfer test (9) becomes:

$$\Delta W = \frac{1}{2(n_Q+1)} \left(-2W_Q + \sum_{x_i \in Q} d(t, x_i) \right) + \frac{1}{2(n_P-1)} \left(2W_P + \sum_{x_i \in P} d(t, x_i) \right) < 0 \tag{12}$$

For each object, there can be more than one cluster that satisfies the transfer test. One way is to pick the first eligible cluster, that is, if clusters are scanned from 1 to K, the one with the smallest index is picked. By computing the transfer test for every cluster, we can also choose the cluster that has the most decrease of the objective function. Although the time cost is higher than the first method, it maybe worthwhile that better optima are achieved [16]. In the experiment section, we will use this second method.

The task remains is the computation of the transfer test (12). For W and n, we can keep them in computer memory, and update them every time a transfer happens. W updates as in (10) and (11), and n updates as to plus one or minus one. Now the update of W and the transfer test requires us to compute $\sum_{x_i \in Q} d(t, x_i)$. For numerical data, the computation has a time complexity of $O(n)$ for each cluster. It's more efficient to compute distance from an object to a center, which is $O(A)$, where A is the number of attributes; that's why we use the center-based procedure in k-means. But for categorical data, we find that it's efficient to compute $\sum_{x_i \in Q} d(t, x_i)$, as long as we maintain a frequency table for every cluster. This quality makes transfer algorithm efficient and suitable for categorical data.

We use the simple matching dissimilarity measure (1) to illustrate: the total dissimilarities between an object t and all objects in a cluster Q is:

$$\sum_{x_i \in Q} d(t, x_i) = \sum_{x_i \in Q} \sum_{j=1}^{A} d_j(t_j, x_{ij}) = \sum_{j=1}^{A} \sum_{x_i \in Q} d_j(t_j, x_{ij}) = \sum_{j=1}^{A} (n_Q - f_j) = n_Q A - \sum_{j=1}^{A} f_j \tag{13}$$

f_j is the frequency of attribute value t_j in cluster Q, read from the frequency table. We maintain a frequency table for every cluster, which contains the frequencies of all attribute values in the cluster. For the first cluster in Example 1, the frequency table is shown in table 2

Table 2. A Particular Frequency Table

Attributes	A1		A2		A3			
Attribute Values	a	b	p	q	x	y	z	w
Frequency	3	2	3	2	2	1	1	1

We see that the time cost of the transfer test is only $O(A)$, making the transfer algorithm efficient for categorical data. For other dissimilarity measures that are based on co-occurrence of categorical values, such as those listed in [9], their transfer tests are also efficient using the same method.

Finally, the algorithm is as follows:

1. Initialize an random partition of the dataset.
2. Scan every object of the dataset. If there is an object that satisfies the transfer test, relocate it to the cluster that has the most decrease of the objective function (7), i.e. the one with the biggest ΔW .Suppose the object is transferred from cluster P to Q. Update W_P, W_Q, n_P, n_Q, and the frequency tables of cluster P and Q.
3. Repeat Step 2 until no objects are transferred in a full cycle scan of the whole data set.

Let k be the number of clusters, A be the number of attributes, n be the number of objects, i be the number of iterations run before convergence. The time complexity is $O(nikA)$, which is the same as k-modes, and lower than the binary method.

4 Experimental Results

4.1 Clustering Efficacy

We have argued that the within-cluster dispersion objective is more informative than center-based objectives of k-modes, thus our algorithm produces more accurate clustering results. Here we show it does produce better results with three benchmark real datasets.

The first dataset is soybean (small) from UCI Machine Learning Repository [18], whose datasets are widely used in the research of clustering categorical data. The soybean dataset has 47 instances, each being described by 35 attributes. Each instance is labeled as one of four diseases: D1, D2, D3, and D4. Except for D4, which has 17 instances, all other diseases have 10 instances each. To reduce the effect of object order, we randomly reorder the objects in each run of the algorithm.

First we show the convergence of our algorithm. In Figure 1, we show 50 curves for 50 runs on the soybean dataset. Each curve plots the change of the objective function values with respect to all the transfers. We can see that the objective function values are decreasing in each curve. This verifies the efficacy of the transfer test (9), whose goal is to make an eligible transfer. We also see that the algorithm stops after a finite number of transfers. The objective function values do not decrease any more, because no more transfers are eligible. Thus our algorithm always converges to local optima.

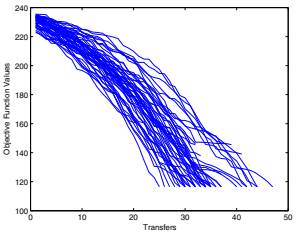


Fig. 1. The Objective Function

	Clus 1	Clus 2	Clus 3	Clus 4
D1		10		
D2			10	
D3	10			
D4		5		12

Fig. 2. Classification Matrix

For each clustering result we use a classification matrix to analyze the correspondence between the clusters and the disease classes of the instances. For example, in Figure 2 five instances from the disease class D4 are incorrectly classified into Cluster 2, while other three disease classes are correctly clustered.

To measure the performance of clustering results, we use the same criterions as in [17]: accuracy (AC), precision (PR), and recall (RE).

$$accuracy = \frac{\sum_{i=1}^{k} a_i}{n} \quad precision = \frac{\sum_{i=1}^{k} \left(\frac{a_i}{a_i+b_i}\right)}{k} \quad recall = \frac{\sum_{j=1}^{k} \left(\frac{a_j}{a_j+c_j}\right)}{k} \tag{14}$$

Where n is the total number of objects, a_i is the number of correctly classified objects in cluster i, which also means the number of objects with class label that dominate cluster i. b_i is the number of objects that are incorrectly classified into cluster i. a_j is the number of objects that are in class j and in a cluster, c_j is the number of objects that are in class j but not in a cluster. For example, in Figure 2, the accuracy is $(10+10+10+12)/47$, the precision is $(10/10+10/15+10/10+12/12)/4$, the recall is $(10/10+10/10+10/10+12/17)/4$. Table 3 shows the results: we run our algorithm 1000 times, while the results of k-modes and k-modes with new dissimilarity measure are from Ng [17]. We can see that our algorithm performs better in every measure, which proves the better efficacy of our objective function.

Table 3. Performance Comparison on Soybean Dataset

	Mean			Standard Deviation		
	Our algorithm	New k-modes	k-modes	Our algorithm	New k-modes	k-modes
AC	0.9510	0.9132	0.8260	0.0879	0.1053	0.1109
PR	0.9704	0.9500	0.8810	0.0537	0.0669	0.0901
RE	0.9721	0.9520	0.8840	0.0507	0.0670	0.0812
	Minimum			Maximum		
	Our algorithm	New k-modes	k-modes	Our algorithm	New k-modes	k-modes
AC	0.7872	0.7760	0.5740	1.0000	1.0000	1.0000
PR	0.8322	0.7824	0.6470	1.0000	1.0000	1.0000
RE	0.8574	0.7361	0.7080	1.0000	1.0000	1.0000

In k-modes [11], the accuracy for the soybean dataset is presented as follows: define clustering results with $accuracy > 87\%$ as *good* results. K-modes produces 64% *good* results. In our algorithm 79.8% results are *perfect* results with $accuracy = 100\%$. An important comment is: the clustering results with 100% accuracy also have the minimum value of the objective function, which illustrates the soundness of the objective function.

Now we show the results from another popular dataset called mushroom dataset [18]. He [12] has the results of clustering accuracy of k-modes and k-modes with new dissimilarity measure. So we will compare the clustering accuracy of the three algorithms. To be consistent with the data in [12], we show the clustering error: $Error = 1 - Accuracy$. The result is in Figure 3. We can

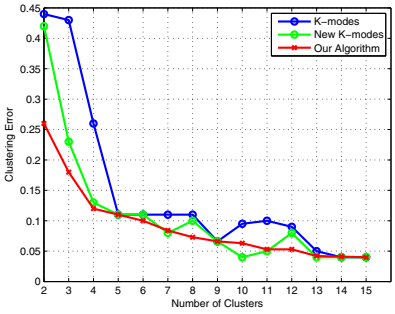


Fig. 3. Clustering Error of the Mushroom dataset

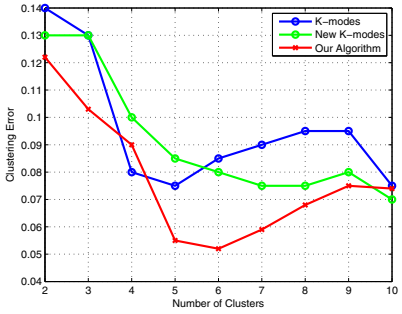


Fig. 4. Clustering Error of the Congressional Votes dataset

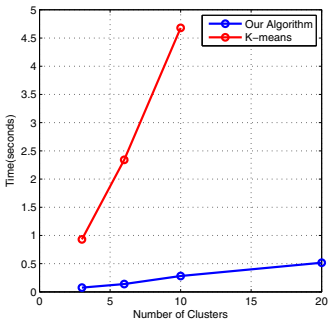
see that our algorithm generally provides better or equal results with respect to different numbers of clusters. When the number of clusters is small, our errors are significantly smaller.

The third dataset is congressional voting records from UCI Machine Learning Repository. The results are also compared with k-modes and k-modes with new dissimilarity measure [12], in Figure 4. We see that in most cases, our algorithm have smaller errors. Only when the number of clusters is 4 or 10, the errors of our algorithm are slightly bigger. So this dataset further proves the superior effectiveness of our objective function.

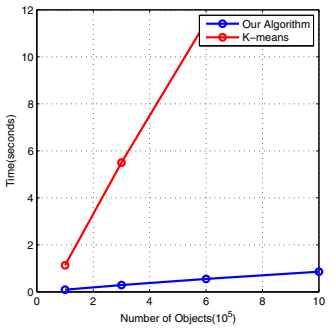
4.2 Scalability

Here we test the scalability of the algorithm using synthetic data [19]. The results are in Figure 5. When one of the four factors is the variable, the other three are fixed, and these fixed values are: number of objects 100000, number of features 10, number of clusters 4, and number of categories 20. We can see that the growth of running time is linear to the number of objects, the number of features and the number of clusters respectively. This is consistent with the complexity analysis in Section 3.

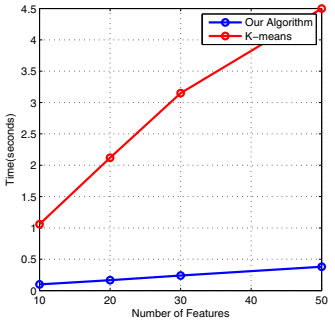
In Figure 5 we also compare the running time with k-means, which first transforms categorical data into binary data. As discussed in the first section, the time complexity of the binary method has a factor of the number of categories. Thus in Figure 5d, we can see the running time grows linearly with respect to the number of categories. While in our algorithm, as the number of categories grows, the running time is constant. Thus our algorithm is more efficient than the binary method. From Figure 5a 5b 5c, we can also see that the running time of our algorithm grows slower than the k-means method. Intuitively it is because the number of transfers in our algorithm is much smaller than the number of times of computing distances between data points and centroids in k-means.



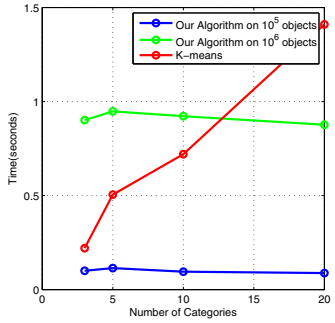
(a) Time vs. Number of Clusters



(b) Time vs. Number of Objects



(c) Time vs. Number of Features



(d) Time vs. Number of Categories

Fig. 5. Scalability Results

5 Conclusions

Existing partition clustering algorithms for categorical data either have efficiency or efficacy weaknesses. We propose an objective function that generalizes from k-means, and demonstrate that it produces more accurate clustering results with real-life datasets. The objective function can also incorporate different dissimilarity measures, which is a good quality for categorical data. At last, due to the nature of categorical data, we can use the transfer algorithm to efficiently optimize the objective. This efficiency is based on storing the frequencies of categorical values. In conclusion, our method is a better clustering algorithm for categorical data in the partition paradigm.

Acknowledgements. This research was partially supported by the National Technology Support Program under grant of 2011BAH16B04, the National Natural Science Foundation of China under grant of No. 61173176.

References

1. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
2. Ganti, V., Gehrke, J., Ramakrishnan, R.: CACTUS–clustering categorical data using summaries. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 73–83. ACM (1999)
3. Gibson, D., Kleinberg, J., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. Databases, 1 (1998)
4. Barbar, D., Li, Y., Couto, J.: COOLCAT: an entropy-based algorithm for categorical clustering. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 582–589. ACM (2002)
5. Andritsos, P., Tsaparas, P., Miller, R.J., Sevcik, K.C.: LIMBO: Scalable clustering of categorical data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004)
6. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366 (2000)
7. Steinley, D.: K–means clustering: A half–century synthesis. British Journal of Mathematical and Statistical Psychology 59(1), 1–34 (2006)
8. Ralambondrainy, H.: A conceptual version of the K-means algorithm. Pattern Recognition Letters 16(11), 1147–1157 (1995)
9. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. Red 30(2), 3 (2008)
10. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications 36(2), 3336–3341 (2009)
11. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2(3), 283–304 (1998)
12. He, Z., Deng, S., Xu, X.: Improving K-modes algorithm considering frequencies of attribute values in mode. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-M., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005, Part I. LNCS (LNAI), vol. 3801, pp. 157–162. Springer, Heidelberg (2005)
13. San, O.M., Huynh, V.N., Nakamori, Y.: An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science 14(2), 241–248 (2004)
14. Bai, L., Liang, J., Dang, C., et al.: The Impact of Cluster Representatives on the Convergence of the K-Modes Type Clustering (2012)
15. Banfield, C.F., Bassill, L.C.: Algorithm AS 113. A transfer algorithm for non-hierarchical classification. Applied Statistics 26, 206–210 (1977)
16. Tarsitano, A.: A computational study of several relocation methods for k-means algorithms. Pattern Recognition 36(12), 2955–2966 (2003)
17. Ng, M.K., Li, M.J., Huang, J.Z., et al.: On the impact of dissimilarity measure in k-modes clustering algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 503–507 (2007)
18. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2013), `http://archive.ics.uci.edu/ml`
19. Gabor Melli. The datgen Dataset Generator, `http://www.datasetgenerator.com`

eDARA: Ensembles DARA

Chung Seng Kheau, Rayner Alfred, and HuiKeng Lau

School of Engineering and Information Technology, Universiti Malaysia Sabah,
Jalan UMS, 88400, Kota Kinabalu, Sabah, Malaysia
{kheau,ralfred,hklau}@ums.edu.my

Abstract. The ever-growing amount of digital data stored in relational databases resulted in the need for new approaches to extract useful information from these databases. One of those approaches, the DARA algorithm, is designed to transform data stored in relational databases into a vector space representation utilising information retrieval theory. The DARA algorithm has shown to produce improvements over other state-of-the-art approaches. However, the DARA suffers a major drawback when the cardinality of attributes in relations are very high. This is because the size of the vector space representation depends on the number of unique values of all attributes in the dataset. This issue can be solved by reducing the number of features generated from the DARA transformation process by selecting only part of the relevant features to be processed. Since relational data is transformed into a vector space representation (in the form of $TF\text{-}IDF$), only numerical values will be used to represent each record. As a result, discretizing these numerical attributes may also reduce the dimensionality of the transformed dataset. When clustering is applied to these datasets, clustering results of various dimensions may be produced as the number of bins used to discretize these numerical attributes is varied. From these clustering results, a final consensus clustering can be applied to produce a single clustering result which is a better fit, in some sense, than the existing clusterings. In this study, an ensemble DARA clustering approach that provides a mechanism to represent the consensus across multiple runs of a clustering algorithm on the relational datasets is proposed.

Keywords: Relational databases, data mining, one-to-many relations, vector space model, ensemble clustering, data summarization.

1 Introduction

Nowadays, most scientific data are digitally stored in multi relational databases. A database consists of a collection of data items that are stored in a set of relations, also known as table. There are many approaches that have been introduced in learning relational data. The Dynamic Aggregation of Relational Attributes (DARA) [1,2] is one of the approaches that was introduced for data summarization (clustering) on the relational data in order to extract useful information from

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 71–82, 2013.

these databases. The DARA approach has been shown to produce better predictive accuracies [1] when compared to other approaches such as the Inductive Logic Programming (ILP)-based [9] and the Relational Instance-Based Learning (RIBL) [10,11] approaches. In the DARA algorithm, a relational dataset is transformed into a propositional dataset containing attribute-value features. Then, this propositional dataset is summarized (i.e. clustered) and the clusters result is embedded into the target table as a new attribute feature before a classification task is performed. For instance, in Fig. 1, a target table is associated with a non-target table with one-to-many relationship. These relationships can be captured by aggregating them into a bag of patterns. When a relational data is transformed into a vector-based data, a clustering task can be performed to summarize this relational data. A new feature, F_{new}, that represents the cluster results is then embedded into the target table. A predictive or descriptive task is then performed on this updated target table.

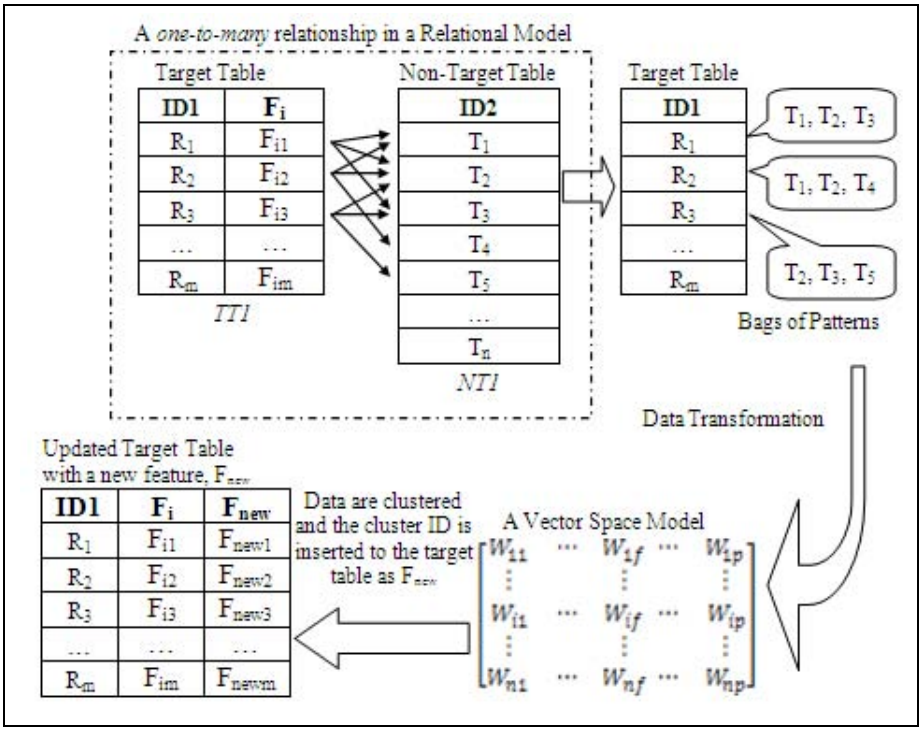


Fig. 1. The Framework of Dynamic Aggregation of Relational Attributes (DARA)

Clustering is a process of grouping a collection of individual objects and the efficiency of the DARA clustering process depends on the parameters of the clustering algorithm and datasets to be processed. These parameters include the number of clusters generated, number of features selected, number of features

constructed, distance methods, seeds' positions for the clusters and the number of bins used to discretize the numerical attributes. Generally, various sets of cluster results or various dimensions of clustering results can be produced by varying the parameter settings. With different dimensions of clusters, a consensus clustering can be applied to produce a single clustering result which is a better fit, in some sense, than the existing clusterings.The original DARA algorithm does not provide a mechanism to reconcile the clustering results of the same dataset from different sources or from different runs of the same algorithm. This is addressed by clustering ensembles [17] that proposes that it would be beneficial to combine the strengths of many different individual clustering algorithms in order to come out an absolute result.

With multiple dimensions of clusterings, clustering ensemble is shown to be able to perform beyond what a single clustering algorithm can achieve in terms of [17,18]: *robustness* - better average performance; *novelty* - finding a combined solution which is unattainable by any single clustering algorithm; *stability and confidence estimation* - lower sensitivity to noise, outliers, or sampling variations; *parallelization and finally scalability* - clustering of data subset can be worked in parallel and subsequent combination of results, and also ability to *integrate solution from multiple distributed sources*. In other words, the clustering ensembles method [15] leverages the consensus across multiple clustering solutions involved and combines them into a single consensus.

In this paper, we propose ensemble DARA (eDARA) framework that provides a robust clustering of data along multiple dimensions (e.g., based on the number of bins used to discretize numerical attributes) and aggregates these different dimensions of clusterings. In other words, eDARA provides a mechanism to represent the consensus across multiple runs of a clustering algorithm on the relational datasets. This is performed in order to determine the best number of clusters in the data, and also to assess the stability of the discovered clusters.

This paper is organized as followed. Section 2 describes some of the works related to relational learning and also ensemble clustering. Section 3 describes the general overview of the DARA algorithm. Section 4 discusses the proposed architecture of ensemble DARA. Section 5 outlines the experimental setup and results and finally Section 6 concludes this paper.

2 Related Works

Clustering Ensembles have been widely used in many real world applications that involve clustering analysis. The main purpose of clustering ensemble is to consolidate all the clustering results generated from different dimensions of clustering methods or datasets used into a single clustering solution. The multiple dimensions of clustering results can be obtained by [20,15]:

1. Applying different clustering algorithms to produce multiple clustering results [24,22,23].
2. Partitioning the original data and then clustering them partially [27,28,21]. The purpose is to handle a large size of data.

3. Selecting different features for subsequent clustering process [19,31].
4. Using a same clustering algorithm with different parameter settings to produce different sets of clusters [25,26,32]. These parameters can be the number of clusters, discretization, and density of clustering solution.

A clustering ensemble can be used to combine several partition of clusters generated from multiple clustering algorithms. Yi *et al* combined several partition of clusters generated from multiple clustering algorithms into a matrix by applying matrix completion algorithms and then the completed matrix will be clustered by using an efficient spectral clustering algorithm [22]. Nguyen and Hiemstra [23] have proposed a method to improve the quality of diversification result, by combining the clusters generated from two clustering methods (LDA and K-means) and also including the clusters produced by two types of data (document text and anchor text).

Partitioning several portions of the whole data is another way in which multiple dimensions of lcuwstering results can be obtained. One of the real-world applications that applies ensemble technique in data modeling is the prediction of yield in river basins. In this application, the ensemble cluster method is applied to combine multiple clustering schemes to produce a better scheme that deliver similar homogeneous basins. The multiple clustering schemes are obtained by partitioning several portions of the whole data, in which each portion data is clustered separately by using the same clustering algorithm and then produces multiple clustering results. The multiple clustering results are then combined via a consensus function in order to get a more robust clustering result [21].

Hong *et al* [31] used ensembles method to combine the clustering results populated by applying feature selection algorithm for unsupervised clustering. This research is mainly conducted to search for a subset of all features that is able to achieve the most similar clustering solution to one that produced by an ensemble learning algorithm.

Hong *et al* [32] proposed Spectral Clustering ensemble (SCE) algorithm that manipulate the parameter settings of a same clustering algorithm. The random of subspace, scaling parameter, and Nystrom approximation are applied to construct the SCE so that it able to produce multiple SC results and then the results are combined to extract better SC final result. In our study, the multiple dimensions of clustering results is obtained by applying different parameter settings on a DARA framework to learn relational database in order to produce different sets of clustering results. This can be seen on our experiments that combining the clustering results produce by different settings of bins into a new clustering result and then proceed to subsequence clustering process.

3 The Framework of Ensemble DARA (eDARA)

The proposed ensemble DARA (eDARA) clustering is an extended work of the DARA approach. In other words, the basic transformation process of DARA algorithm is still applied in eDARA. However, multiple dimensions of clusterings

are performed in the ensemble DARA (eDARA) by clustering m different sets of transformed data that are obtained from the same data source by varying the number of bins when discretizing the numerical attributes. Figure 2 illustrates the proposed framework of the eDARA approach. There are three main phases in eDARA framework, namely configuration phase, consensus phase and characterization phase.

In the configuration phase, m different dimensions of the same data are clustered by using the same clustering algorithm. These m different dimensions of the same data are obtained by discretizing the numerical values for all features selected by using different number of bins. In this work, all numerical values are discretized according to predefined number of bins (e.g., bin = 3, 5, 7 and 9). In the consensus phase, a consensus function is used to reconcile all the m clustering results obtained from the configuration phase in order to produce a single set of clustering result. Finally, in the characterization phase, the final clustering result will be embedded into the initial target table and further data modelling task can be performed on this newly updated target table.

3.1 Configuration Phase

The configuration phase consists of three major stages: Data Preparation Stage; Data Transformation Stage; Data Summarization Stage.

Data Preparation Stage. In data preparation, the dimensionality of the relational datasets can be reduced via discretization of continuous attribute values [3], feature selection or feature construction [2]. The performance of the DARA transformation process can be improved by reducing the dimensionality of the relational datasets [your previous work].

Data Transformation Stage. In this stage, the DARA algorithm is used to transform a relational data into a propositional dataset containing attribute-value features. In relational databases, a single record stored in a target table may be associated with multiple records stored in a non-target table. To learn this relational data, a data transformation process is performed on this relational data by transforming the data into a vector space representation, i.e., transformed into a $TF\text{-}IDF$ weighted frequency matrix (vector space model) [14]. In a vector space model, a row of data is considered as a record that is represented by a bag of patterns and each record is differentiated by using the frequency of each pattern that exists in the dataset. For instance, in a relational database, the target table (labeled data) is normally linked to other non-target tables with one-to-many relationships. By using the vector space model, a $TF\text{-}IDF$ weighted frequency matrix is generated to represent the relational dataset. The rows of the $TF\text{-}IDF$ weighted frequency matrix represent the instances of the target table and the columns represent the features of each instance that are weighted using the $TF\text{-}IDF$ weighted frequency value, W_i, which is computed using Equation 1,

$$W_i = f_i \cdot \log(D/d_i) \tag{1}$$

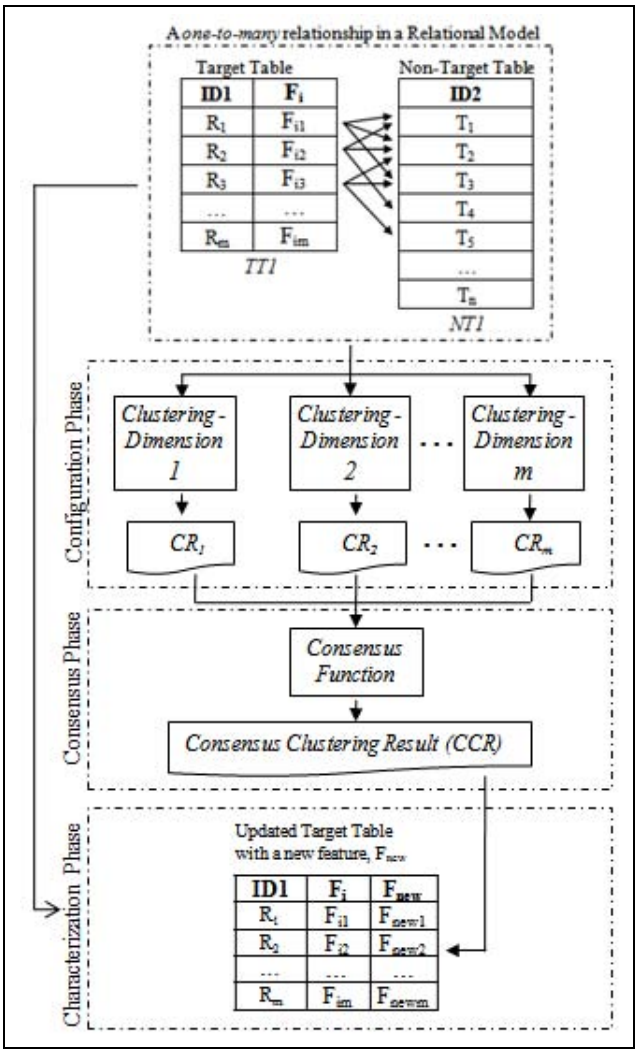


Fig. 2. Ensembles DARA Architecture

where f_i is the frequency of the ith feature in the dataset, D is the total number of instances, d_i is the number of instances containing the ith feature. Since all features in the $TF\text{-}IDF$ weighted frequency matrix contain numerical values, several different dimensions of the same data can be produced by discretizing these numerical values using different number of bins, b. In this work, these continuous features are discretized by using the Equal-Width discretization method with $b = 3$, 5, 7 and 9. This discretization method divides the range of observed values for a particular feature into b equal sized bins, where b is a parameter provided by the user. The interval (Equation 2) can be calculated based on the

value of b parameter supplied by the user, and the minimum V_{min}, and maximum, V_{max} values found by shorting the observed values of a feature,

$$interval = (V_{max} - V_{min})/b \tag{2}$$

and then the boundaries can be computed using Equation 3, where i=1,. . ., b-1.

$$boundaries = V_{min} + (i \times interval) \tag{3}$$

In addition to that, a feature selection method can also be used to reduce the number of features selected before the clustering process can be conducted [4]. In this work, all the discretized features or attributes will be scored. Given a particular feature F, the information gain for this feature, denoted $InfoGain(F)$, represents the difference between the class entropy in data set before the usage of feature F, denoted $Ent(C)$, and the usage of feature F with splitting the data set into subsets, denoted $Ent(C/F)$, as presented in Equation 4.

$$InfoGain(F) = Ent(C) - Ent(C/F) \tag{4}$$

where

$$Ent(C) = -\sum_{j=1}^{n} Pr(C_j) \cdot \log_2 Pr(C_j) \tag{5}$$

and

$$Ent(C/F) = -\sum_{i=0}^{n} Pr(F_i) \cdot (-\sum_{i=0}^{n} Pr(C/F_i) \cdot \log_2 Pr(C_j/F_i)) \tag{6}$$

Data Summarization Stage In this stage, the k-means algorithm is used to cluster m different dimensions of the same data that are obtained by discretizing the selected features that contain continuous values by using different number of bins. This partitional clustering method is based on an iterative relocation process that partitions a set of dataset points into a specified number of clusters. In this work, the k initial centers are randomly selected.

3.2 Consensus Phase

In the consensus phase, there are several types of functions that can be used as a consensus function such as Hypergraph Partitioning [33,15,19], Voting Approach [18,24,20,25], Mutual Information Algorithm [18,20], Co-association based functions [20,35,36] and Finite Mixture model [18,34]. In this work, we apply a consensus clustering [15] in order to find the best single clustering result. Before the consensus clustering process can be performed, all the m clustering results obtained from the configuration phase (see Figure 2) will be consolidated. The process of consolidating all the clustering results is illustrated in Fig. 3, where CR_1, CR_2 and CR_m are the clustering results obtained from the configuration phase. After the consolidation, a new consolidated clustering result, CCR is produced which consists of all the m clustering results. A consensus clustering is then performed on the consolidated clustering results to find a single clustering result, CR_s.

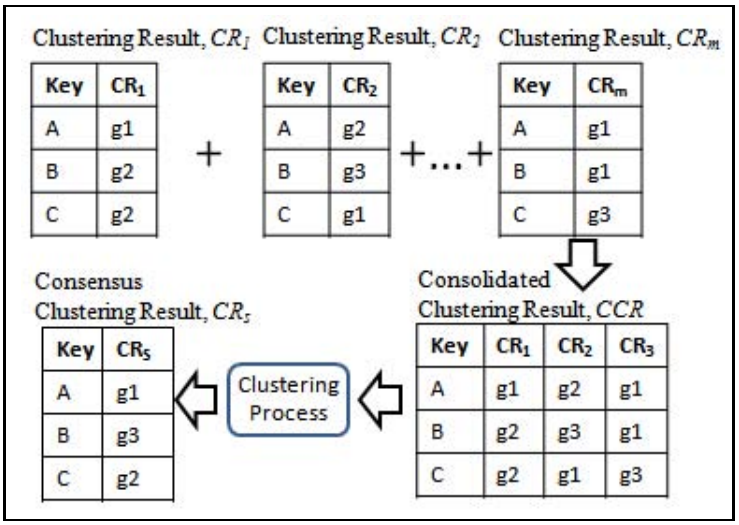


Fig. 3. Consolidating all the m clustering results for the subsequence consensus clustering process

3.3 Characterization Phase

In this phase, further data modeling process can be conducted (e.g., using $WEKA$ [16]) to extract useful information from the updated target table.

4 Experimental Design and Results

A series of experiments has been conducted to observe the influence of applying consensus clustering to the DARA algorithm for a classification task. The summarized or clustered data are coupled with the C4.5 classifier (from WEKA) [16], as the clustering algorithms on the eDARAs. The effectiveness of the data transformation with respect to C4.5 is evaluated. The C4.5 learning algorithm [29] is a state-of-the-art top-down method for inducing decision trees. The datasets used is the mutagenesis dataset (B1, B2 and B3) [30] and the performance accuracy is computed based on 10-fold cross-validation procedure.

In the experiments, all continuous attributes are discretized first prior to the learning process. Since the vector space model (e.g., $TF\text{-}IDF$) is in a numerical data type and this may cause the range of values to be very large, all continuous values in the vector space model are discretized. The number of bins used to discretize the continuous features is 3, 5, 7 and 9.

Next, a feature selection process [your previous work] is performed to reduce the dimensionality of the vector space model of the dataset. The scoring of the given feature, F, is computed based on the information gain, $InfoGain(F)$ as shown in Equation 4. The scores of all these features are used to rank the features. The number of features selected for the data summarization is based

Table 1. Performance accuracy (%) of 10-fold cross-validation of C4.5 on Feature Selection, Bin, and Combined Bin on Mutagenesis datasets (B1, B2, and B3)

Mutagenesis Datasets	# of Bin	Without Feature Selection	S, Feature Selection (%)							
			90	80	70	60	50	40	30	20
B1	3	82.5	80.3	81.9	80.9	82.5	83.5	81.4	81.4	80.9
	5	84.6	80.9	84.0	80.9	80.9	81.4	81.4	80.9	80.9
	7	82.5	80.9	83.0	79.9	79.3	82.0	81.4	80.9	80.9
	9	84.6	84.0	80.9	79.9	81.4	80.9	81.4	80.9	80.9
	3+5+7+9	**85.1**	81.4	83.5	**81.4**	**82.5**	81.9	**81.4**	**81.9**	**80.9**
B2	3	84.6	82.5	84.6	84.6	82.5	79.8	81.4	82.5	80.3
	5	84.6	84.6	84.6	84.6	84.0	79.8	81.4	84.0	80.9
	7	84.6	83.0	84.6	84.6	83.0	79.8	81.4	84.0	79.3
	9	83.0	84.6	83.0	84.6	83.5	80.3	84.0	84.0	80.3
	3+5+7+9	**84.6**	**84.6**	83.5	**84.6**	83.5	**80.3**	**85.1**	83.5	**80.9**
B3	3	82.5	79.8	79.8	79.8	80.3	80.3	81.9	81.4	81.4
	5	83.0	84.6	80.3	81.9	81.4	81.4	81.4	80.9	80.9
	7	84.0	83.5	78.7	81.9	81.9	81.4	81.4	80.3	80.9
	9	83.0	83.5	80.3	79.8	80.9	80.9	81.4	81.4	80.9
	3+5+7+9	82.5	**85.1**	**81.4**	79.8	**82.5**	**82.0**	**81.4**	**81.4**	**80.9**

on the threshold, t, which determines the percentage of features to be selected for the data summarization process. For instance, if the number of feature is n and the threshold t is 0.8, only 80 percent of n features will be involved in the data summarization process. In this experiment, the percentage of features selected ranges from 20 to 100 (e.g., 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0). The DARA algorithm will cluster the discretized data based on the selected features accordingly.

Finally, a consensus clustering is performed to reconcile the clustering results of the same data set from different runs of the same algorithm with different parameter settings (i.e., number of bins used to discretize numerical values). Here, all the clustering results obtained from all bins with the same percentage of features selected are combined into a single dataset and then forwarded to the consensus clustering process in order to produce a final clustering result.

Table 1 shows the predictive accuracy results of the C4.5 classifier on the Mutagenesis datasets (B1, B2, and B3). These predictive accuracies are based on the number of bins used, the percentage of features involved in the data summarization process, and also the combined clustering results in the consensus clustering process. For B1 dataset, the results indicate that the consensus clustering (i.e., Bin(3+5+7+9)) provides a better or comparable predictive accuracy result for 6 out of 9 settings. For instance, the predictive accuracies for bin(3+5+7+9) on the selected features increase to 85.1% for the 100% features selected (without feature selection), 81.4% for the 70% features selected and 81.9% for the 30% features selected. As for the rest, the predictive accuracies for bin(3+5+7+9) on the selected features remained the same.

Similarly, the results indicate that the consensus clustering (e.g., Bin(3+5+7+9)) provides a better or comparable predictive accuracy result for 6 out of 9 settings for the B2 dataset. For instance, the predictive accuracies for bin(3+5+7+9) on the selected features increase to 85.1% for the 40% features selected. The predictive accuracies for bin(3+5+7+9) on the selected features remained the same for the dataset with 100%, 90%, 70%, 60%, 50% and 20% features selected.

Finally for the B3 dataset, the consensus clustering provides a better or comparable predictive accuracy result for 5 out of 9 settings. The predictive accuracies for bin(3+5+7+9) on the selected features increase to 85.1% for the 90% feature selected, 81.4% for the 80% features selected features, 82.5% for the 60% features selected and 82.0% for the 50% features selected. As for other percentage of selected features, the predictive accuracies for bin(3+5+7+9) remained the same.

5 Conclusion

In this study, an ensemble DARA (eDARA) clustering approach is proposed. The proposed eDARA framework provides a mechanism to find the best clustering result after reconciling several dimensions of clustering results obtained by manipulating the number of bins used to discretize the DARA transformed data. The experimental results have shown that the consensus clustering in eDARA could improve the predictive accuracy in a classification task in learning relational data. In the near future, other possible investigation of the eDARA such as the experiments with different clustering algorithms to produce multiple clustering results, and using the same clustering algorithm but different sets of clusters will be investigated.

References

1. Alfred, R.: The Study of Dynamic Aggregation of Relational Attributes on Relational Data Mining. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 214–226. Springer, Heidelberg (2007)
2. Alfred, R.: Optimizing feature construction process for dynamic aggregation of relational attributes. J. Comput. Sci. 5, 864–877 (2009), doi:10.3844/jcssp.2009.864.877
3. Alfred, R., Kazakov, D.: Discretization Numbers for Multiple-Instances Problem in Relational Database. In: Ioannidis, Y., Novikov, B., Rachev, B. (eds.) ADBIS 2007. LNCS, vol. 4690, pp. 55–65. Springer, Heidelberg (2007)
4. Kheau, C.S., Alfred, R., Keng, L.H.: Dimensionality reduction in data summarization approach to learning relational data. In: Selamat, A., Nguyen, N.T., Haron, H. (eds.) ACIIDS 2013, Part I. LNCS, vol. 7802, pp. 166–175. Springer, Heidelberg (2013)
5. Karunaratne, T., Bostrom, H., Norinder, U.: Pre-Processing Structured Data for Standard Machine Learning Algorithms by Supervised Graph Propositionalization – a Case Study with Medicinal Chemistry Datasets. In: Ninth International Conference on Machine Learning and Applications, pp. 828–833 (2010)

6. Li, Y., Luan, L., Sheng, Y., Yuan, Y.: Multi-relational Classification Based on the Contribution of Tables. In: International Conference on Artificial Intelligence and Computational Intelligence, pp. 370–374 (2009)
7. Pan, C., Wang, H.-Y.: Multi-relational Classification on the Basic of the Attribute Reduction Twice. Communication and Computer 6(11), 49–52 (2009)
8. He, J., Liu, H., Hu, B., Du, X., Wang, P.: Selecting Effective Features and Relations For Efficient Multi-Relational Classification. Computational Intelligence 26(3), 1467–8640 (2010)
9. Wrobel, S.: Inductive Logic Programming for Knowledge Discovery in Databases: Relational Data Mining, pp. 74–101. Springer, Berlin (2001)
10. Emce, W., Wettschereck, D.: Relational instance-based learning. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 122–130. Morgan Kaufmann, San Matco (1996)
11. Kirsten, M., Wrobel, S., Horvath, T.: Relational Distance Based Clustering. In: Page, D.L. (ed.) ILP 1998. LNCS, vol. 1446, pp. 261–270. Springer, Heidelberg (1998)
12. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and Unsupervised Discretisation of Continuous Features. In: ICML, pp. 194–202 (1995)
13. Knobbe, A.J., de Haas, M., Siebes, A.: Propositionalisation and Aggregates. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 277–288. Springer, Heidelberg (2001)
14. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1984)
15. Strehl, A., Ghosh, J.: Cluster Ensembles A Knowledge Reuse Framework for Combining Multiple Partitions. Journal Machine Learning Resarch, 583–617 (February 2002)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
17. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: Model of consensus and weak partitions. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(12), 1866–1881 (2005)
18. Topchy, A., Jain, A.K., Punch, W.: A mixture model for clustering ensembles. In: SIAM International Conference on Data Mining, Michigan State University (2004)
19. Topchy, A., Minaei Bidgoli, B., Jain, A.K., Punch, W.: Adaptive clustering ensembles. In: Proceeding International Conference on Pattern Recognition (ICPR), Cambridge, UK, pp. 272–275 (2004)
20. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: Proceeding of the Third IEEE International Conference on Data Mining (2003)
21. Ahuja, S.: Regionalization of River Basins Using Cluster Ensemble. Journal of Water Resource and Protection, 560–566 (2012)
22. Yi, J., Yang, T., Jin, R., Jain, A.K., Mahdavi, M.: Robust Ensemble Clustering By Matrix Completion. In: IEEE 12th International Conference on Data Mining, pp. 1176–1181 (2012)
23. Nguyen, D.P., Hiemstra, D.: Ensemble clustering for result diversification. NIST Special Publications (2012)
24. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics Oxford University 19(9), 1090–1099 (2003)
25. Gablentz, W., Koppen, M.: Robust clustering by evolutionary computation. In: Proceeding of the Fifth Online World Conference Soft Computing in Industrial Applications, WSC5 (2000)

26. Luo, H., Jing, F., Xie, X.: Combining multiple clusterings using information theory based genetic algorithm. In: IEEE International Conference on Computational Intelligence and Security, vol. 1, pp. 84–89 (2006)
27. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the 21st International Conference on Machine Learning, Canada (2004)
28. Hong, Y., Kwong, S., Chang, Y., Ren, Q.: Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recognition Society 41(9), 2742–2756 (2008)
29. Quinlan, R.J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Series in Machine Learning (January 1993)
30. Srinivasan, A., Muggleton, S., Sternberg, M.J.E., King, R.D.: Theories for Mutagenicity: A Study in First-Order and Feature-Based Induction. Artificial Intelligence 85(1-2), 277–299 (1996)
31. Hong, Y., Kwong, S., Chang, Y., Ren, Q.: Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recognition 41(9), 2742–2756 (2008)
32. Zhang, X., Jiao, L., Liu, F., Bo, L., Gong, M.: Spectral Clustering Ensemble Applied to SAR Image Segmentation. IEEE Transactions on Geoscience and Remote Sensing 46(7) (July 2008)
33. Karypis, G., Kumar, V.: Solving cluster ensemble problems by correlation's matrix & GA. VLSI Design 11(3), 285–300 (2000)
34. Analoui, M., Sadighian, N.: Multilevel k-way Hypergraph Partitioning. IFIP International Federation for Information Processing, vol. 228, pp. 227–231 (2006)
35. Fred, A.L.N., Jain, A.K.: Data clustering using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 835–850 (2002)
36. Tayanov, V.: Some questions of consensus building using co-association. In: Proceedings of the 11th WSEAS international conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED 2012, pp. 61–66 (2012)

Efficient Mining Maximal Variant and Low Usage Rate Biclusters without Candidate Maintenance in Real Function-Resource Matrix: The DeCluster Algorithm*

Lihua Zhang[1,2], Miao Wang[2,**] , Zhengjun Zhai[1], and Guoqing Wang[1,2]

[1] School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China, 710072
[2] Science and Technology on Avionics Integration Laboratory, Shanghai, China, 200233
riyushui@gmail.com

Abstract. The health of the functional layer includes the status of functional components in the hierarchy range and overall health status of the whole functional layer. This paper proposed an efficient bicluster mining algorithm: *DeCluster*, to effectively mine all biclusters with maximal variant usage rate and maximal low usage rate in the real-valued function-resource matrix. First, a sample weighted graph is constructed, it includes all resource collections between both samples that meet the definition of variant usage rate or low usage rate; then, all biclusters with maximal variant usage rate and low usage rate meeting the definition are mined with the mining method of using depth-first sample-growth in the weight graph made. To improve the mining efficiency of the algorithm, *DeCluster* algorithm uses multiple pruning strategies to ensure the mining of maximal bicluster without candidate maintenance. The experimental result show our algorithm is efficiently than traditional algorithm.

Keywords: bicluster, variant usage rate, low usage rate, function-resource.

1 Introduction

The health of the functional layer includes the status of functional components in the hierarchy range and overall health status of the whole functional layer. Health management objective of the functional layer is the effectiveness of the functional components and the hierarchy and to form function self-organizing platform based on the effectives of functional components. Although studying the effectiveness degree of resources is the base to construct a prognostics and health management system[1]. The health degree of resources directly influences functional health. So, analysis of the call relation of functions and resources can excavate the health relation between resources and functions so as to complete the functions through using healthy resources and improve the health degree of functions. The relation between functions

* Supported by National Key Basic Research Program of China(Grant No. 2014CB744900).
** Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 83–94, 2013.

and resources is denoted as a matrix, where each row represents a resource and each column represents a function, the value in the matrix is the use degree of a function to a resource. This value is defined during functional design, i.e. resource dependence degree of this function in aircraft system in order to complete a function. Through function-resource matrix mining, in order to achieve a group of functions, the resources which can meet all functional demands simultaneously and the resources which can satisfy all functional demands through multiple accesses can be mined, i.e. mine bicluster with variant usage rate or low usage rate from function-resource matrix.

The above mining thought complies with the mining thought of bicluster in data mining field. Biclustering concept was first put forward by Cheng and Church [2]. As a special clustering method, bicluster does not generate cluster in overall experimental conditions, but only finds out the item sets with special significance for specific matrix sample. Thus, biclustering algorithm can mine bicluster with variant usage rate and low usage rate described above from function-resource matrix. Bicluster can be classified into four categories: (i) constant value biclusters, (ii) constant row or column biclusters, (iii) biclusters with coherent values, where each row and column is obtained by addition or multiplication of the previous row and column by a constant value. Currently, large quantities of algorithms based on greedy strategy or exploratory strategy are applied in mining bicluster. Cheng and Church put forward an algorithm based on greedy strategy [2]. This algorithm adopts a low square root residue to delete redundant nodes step by step. Many algorithms based on greedy strategy were proposed [3-5]. RAP algorithm [6] proposed by Kumar et al. can directly mine constant column bicluster from actual chip data in terms of the range support. The ability of item-growth extension and sample-growth at the same time is the most significant advantage of biclustering. Another advantage is that it can be directly used for original data without data standardization [7]. However, biclustering still has some drawbacks [8]. First, bicluster is a NP-hard problem [9]; second, while processing original data, bicluster needs to solve the problem of sensitivity of original data of gene chip to noise; moreover, bicluster algorithm should allow to mine the overlapped clusters, which increases the computation complexity of biclustering algorithm; finally, as biclustering algorithm directly processes original data, it should have a very strong flexibility for different types of bicluster. Therefore, the design of high-efficiency bicluster mining algorithm is a research hotspot currently.

The author has learnt that no biclustering algorithm can mine a bicluster with variant usage rate and one with low usage rate at the same time currently. Therefore, this paper puts forward a new bicluster mining algorithm: *DeCluster* algorithm to effectively mine all biclusters with maximal variant usage rate and maximal low usage rate from the function-resource matrix of true value. As the number of functions is far lower than that of resources in function-resource matrix, this algorithm uses sample-growth for mining. First, a sample weighted graph is constructed, which includes all resource collections between both samples that meet the definition of variant usage rate or low usage rate; then, all biclusters with maximal variant usage rate and low usage rate meeting the definition are mined with the mining method of using

depth-first sample-growth in the weight graph. To improve the mining efficiency of the algorithm, *DeCluster* algorithm uses multiple pruning strategies to ensure the mining of maximal bicluster without candidate maintenance.

2 Problem Description

Function-resource matrix is defined as a two-dimensional real matrix $D = R \times F$, in which row set R represents the set of resources and column set F refers to the set of functions. Element D_{ij} of matrix D is a real number which represents the ability validity or usage rate of resource i supporting function j. $|R|$ is the number of resources in data set D and $|F|$ is the number of functions in data set D. For the convenience of mining, the domain of definition of the original effective value in resource effectiveness matrix is [0,1], where '0' means that this resource is not required during the implementation of some function; '1' means that this resource must be used during the implementation of some function, as shown in table 1.

Table 1. An example of function-resource matrix

	F_1	F_2	F_3	F_4	F_5
R_1	0.8	0.1	0.12	0.09	0.9
R_2	0.2	0.9	0.19	0.21	1
R_3	0.9	0.3	0.29	0.28	0.55
R_4	0.58	1	0.2	0.21	0.9

Table 2. An example of variant usage rate

	F_1	F_2
R_1	0.8	0
R_2	0. 8	0.1
R_3	0.5	0
R_4	0	0.1

Table 3. An example of non-variant usage rate

	F_1	F_2
R_1	0.8	0
R_2	0. 8	0.7
R_3	0.5	0
R_4	0	0.1

Table 4. An example of low usage rate

	F_1	F_2
R_1	0.8	0
R_2	0. 2	0.1
R_3	0.5	0
R_4	0	0.1

The significance of bicluster to be mined from function-resource matrix as shown in Table 1 is to mine a group of functions executed; under this group of functions, the usage rate of the resource is the maximal, i.e. which resources can reach the maximal usage rate when used together. In other words, the resources have the highest effectives when all functions are executed. For example, for a group of functions F_1F_2 (F_1=>$R_1R_2R_3$, F_2=>R_2R_4), these three functions may be called simultaneously. For resource R_2, there are three situations for supporting F_1 and F_2. (1) for F_1, the usage rate of R_2 is high, while it is low for F_1, as shown in Table 2; (2) for both F_1 and F_2, the usage rate of R_2 is high, as shown in Table 3; (3) for both F_1 and F_2, the usage rate of R_2 is low, as shown in Table 4, the health degree in the first and the third conditions is higher than the second condition. This is because in the first and the third conditions resource R_2 can serve F_1 and F_2 at the same time. In the third condition, resource R_2 needs to serve the two functions respectively. This paper puts forward that bicluster mined by *DeCluster* algorithm aims at the first and third conditions. We will give definitions of low usage rate and variant usage rate of resources in real data below:

Definition 1. D is a function-resource usage rate matrix; α is a user-defined parameter used for measuring the degree of association of functions in resources; β is a parameter restricting low usage rate of resources; r is any resource in function-resource usage rate matrix D; F_1 and F_2 are any two functions in D; r should meet the following conditions for relevance in F_1 and F_2 $[\ \forall r \in R \mid (\max_{f\in\{F_1,F_2\}} D_{r,f} - \min_{f\in\{F_1,F_2\}} D_{r,f}) \le \alpha(\min_{f\in\{F_1,F_2\}} |D_{r,f}|)$ and $\max_{f\in\{F_1,F_2\}} D_{r,f} \le \beta]$. If all resources and functions meet the conditions above in a bicluster, this bicluster is one with low usage rate of resources.

It can be obtained from the description in definition 1 that α and β are used to restrict resources with a low usage rate producing each function, e.g. bicluster $F_2F_3F_4(R_1R_3)$ in table 1.

Definition 2. D is a function-resource usage rate matrix; γ is a user-defined parameter used for measuring the variant usage rate of functions in resources; β is a parameter restricting low usage rate of resources; r is any resource in function-resource usage rate matrix D; F_1 and F_2 are two functions in D; r should meet the following conditions for variant usability in F_1 and F_2 $\max_{f\in\{F_1,F_2\}} D_{r,f} \ge \beta$ and $\frac{\max_{f\in\{F_1,F_2\}} D_{r,f}}{\min_{f\in\{F_1,F_2\}} D_{r,f}} \ge \gamma$. If at least one resource in a bicluster meets the conditions above under two functions and meanwhile this resource meets the conditions in definition 1 under other functions, this bicluster is one with variant usage rate of resources.

It can be obtained from the description in definition 2 that at least one resource in bicluster of variant usage rate of resources meets the conditions in the formula in definition 2 under two functions and meanwhile this resource meets the conditions in the formula in definition 1 under other functions. For the convenience of description, such resources are defined as resources with variant usage rate as below:

Definition 3. D is a function-resource usage rate matrix; γ is a user-defined parameter used for measuring the variant usage rate of functions in resources; β is a parameter restricting low usage rate of resources; α is a user-defined parameter used for measuring the degree of association of functions in resources; r is any resource in function-resource usage rate matrix D; F is function set in D and r should meet the following conditions for resources with variant usage rate under F: $\max_{f\in F} D_{r,f} \geq \beta$ and $\frac{\max_{f\in F} D_{r,f}}{\max_{f\in F} 2\, D_{r,f}} \geq \gamma$ and $[\ \forall r \in R \mid (\max_{f\in\{F_1,F_2\}} 2\, D_{r,f} - \min_{f\in\{F_1,F_2\}} D_{r,f}) \leq$ $\leq \alpha(\min_{f\in\{F_1,F_2\}} \left|D_{r,f}\right|)$ and $\max_{f\in\{F_1,F_2\}} 2\, D_{r,f} \leq \beta]$, under which *max* refers to maximal value, *min* refers to minimum value and *max2* refers to the second maximal value.

Therefore, resources in bicluster with variant usage rate of resources must be those meeting definitions 1 and 3. We will define the relationship among resources. The relationship among resources in true data and that among resources in discrete data have the same form of expression and only minor differences in definition.

Definition 4. Assuming that true usage rate values of resource R_1 under functions F_1 and F_2 are V_1 and V_2, R_1 has the following four forms of expression under F_1 and F_2: (1) if V_1 and V_2 meet definition 2 and $V_1 \geq V_2$, the contribution rate of R_1 to F_1 and F_2 meets the requirement of variance and it is expressed as 'R_1'; (2) if V_1 and V_2 meet definition 2 and $V_2 \geq V_1$, the contribution rate of R_1 to F_1 and F_2 meets the requirement of variance and it is expressed as '$*R_1$'; (3) if V_1 and V_2 meet definition 1, the contribution rate of R_1 to F_1 and F_2 meets the requirement of low usage rate and it is expressed as '$-R_1$'; (4) if V_1 and V_2 do not meet definition 1 or 2, they are not recorded.

Therefore, each resource in bicluster mined with *DeCluster* algorithm meets the first or second condition in definition 4 under all functions. To improve the mining efficiency of the algorithm, *DeCluster* algorithm mines biclusters with maximal variant usage rate and maximal low usage rate from function-resource matrix of true value by using column extension without candidate maintenance. The mining process of this algorithm will be introduced in detail in the next section.

3 DeCluster Algorithm

3.1 Construct Sample Relational Weighted Graph

The method of mining modes with sample relational weighted graph was used in *MicroCluster* algorithm[5] to mine bicluster at the earliest. Then, Wang et al. [8, 10] also used sample relational weighted graph to mine bicluster and fault-tolerant bicluster. *DeCluster* algorithm in this paper will adopt undirected sample relational weighted graph (hereinafter referred to as sample weighted graph) to mine biclusters with maximal variant usage rate and maximal low usage rate.

Definition 5. Sample weighted graph can be denoted as the set $G = \{E, V, W\}$. Each node in the node set V in the weighted graph represents a function. If an edge exists between a pair of nodes, this means the resource with variant usage rate or low usage

rate exists below two functions represented by this pair of nodes. The set of the edges is expressed as *E*. The weights of each edge are the resource set meeting the definition of variant usage rate or the definition of low usage rate under the two functions connected with this edge. The set of the weights is expressed as *W*.

According to the description in Definition 1, when the resources among functions satisfy the definition of variant usage rate, the weight between two functions does not meet commutativity. For instance, the weight under F_1F_2 is $R_1{*}R_2R_3$, while the weight under F_2F_1 is $R_1{*}R_2R_3\text{-}R_5$. So, in Definition 5, the weight of each edge is the weight under F_iF_j, where $i<j$. Fig.1 shows weight relationship graph corresponding to Table 1.

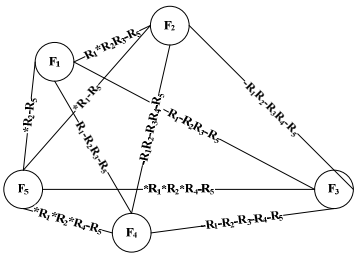


Fig. 1. The sample weighted graph constructed from Table 1

3.2 Mining Maximal Bicluster

After the sample relational weighted graph is made, this section will introduce how *DeCluster* algorithm mines all biclusters with maximal variant usage rate and maximal low usage rate from sample relational weighted graph without candidate maintenance in detail. According to the description in definition 4, biclusters with variant usage rate and low usage rate extended meet anti-monotonicity, i.e. if the bicluster obtained by extension of $F_1F_2\ldots F_n$ does not meet constraint conditions, neither does any superset $F_1F_2\ldots F_nF_m$. Therefore, biclusers with a greater scale can be obtained by extension of the weight on each edge in the weight graph in terms of intersection. However, according to descriptions in definitions 1 and 2, when a new function is introduced in bicluster, it is necessary to calculate the intersection of all edges of the function newly introduced and the resource collection of bicluster extended, thus ensuring that the resource collection under the function newly introduced and that under existing functions meet constraint conditions in definition 1 or 2. For the convenience of design of pruning strategies, it is required to not only calculate the intersection of resources, but also consider symbols before resources during sample-growth and the calculation of intersection of weight, i.e. symbols before resources are also required for 'intersection' calculation. Operational rules of these symbols can be obtained from the definition below.

Definition 6. According to descriptions in definitions 3 and 4, for resource R_1, intersection operational rules of its form of expression are as follows: (1) intersection of 'R_1' and 'R_1' is 'R_1'; (2) intersection of '$-R_1$' and '$-R_1$' is '$-R_1$'; (3) intersection of

'$*R_1$' and '$-R_1$' is '$*R_1$'; (4) intersection of 'R_1' and '$-R_1$' is 'R_1'; (5) intersection of '$*R_1$' and 'R_1' is '$*R_1$'.

It can be seen from definition 3 that the calculation of intersection of 'R_1' and '$*R_1$' will not occur. Therefore, its rules are not provided in definition 6. During function extension, with the increase of functions, the calculation of intersection of multiple forms of expression of the same resource will occur. The intersection can be calculated according to operational rules described in definition 6 according to the sequence. We will introduce how *DeCluster* algorithm uses pruning strategies to mine all biclusters with maximal variant usage rate and maximal low usage rate from sample relationship weight graph without candidate maintenance in detail. This paper will judge maximal bicluster with the method of prior detection put forward in [11] without candidate maintenance. That is to say, if resources under the current candidate sample and some prior candidate sample (mined sample) have some inclusion relation, i.e. all biclusters produced by the current candidate sample can be produced by some prior candidate sample, the current candidate sample can be pruned. During the pruning design of backward checking, if F_1 is the prior candidate function of F_2, the weight on two function edges is the resource collection information of F_2F_1rather than F_1F_2. As resources under F_1F_2 and F_2F_1 have different forms of expression, the sample weighted graph made by this algorithm is a directed graph rather than undirected graph. For F_n and F_m, it is necessary to build edges on F_nF_m and F_mF_n respectively. However, for F_nF_m and F_mF_n, the difference of weights on the edge is the interchange of resource expression forms 'R_1' and '$*R_1$'. Therefore, for saving the storage space, the storage of weight is only that of weight on F_iF_{i+1}edge. The weight on $F_{i+1}F_i$ edge can be calculated with F_iF_{i+1}.

Resource R_1 is respectively expressed as 'R_1' and '$*R_1$' above when the form of expression of resources is illustrated, just for the convenience of design of pruning strategies. If a resource in the current candidate function to be extended meets the form of 'R_1', this resource can be pruned according to the lemma below.

Lemma 1. Assuming that P is the bicluster with variant usage rate to be extended currently; M is the candidate function set of P and N is the prior candidate function set of P. If the form of expression is 'R_j' for any resource R_j in candidate function $M_i(M_i \in M)$ and there is a prior candidate function $N_j(N_j \in N)$ under which resource R_j also exists and resource R_j must exist in PN_jM_p and PN_jM_i for other candidate samples $M_p$in M, resource $R_j$in M_i can be obtained by extension of prior candidate function N_j.

If a resource in the current candidate function to be extended meets the form of '$*R_1$', it should be judged whether this resource can be pruned according to the weight of prior candidate function. Therefore, the following theorem can be used for pruning.

Lemma 2. Assuming that P is the bicluster with variant usage rate to be extended currently; M is the candidate function set of P and N is the prior candidate function set of P. If the form of expression is '$*R_j$' for any resource R_j in candidate function $M_i(M_i \in M)$ and there is a prior candidate function $N_j(N_j \in N)$ under which resource R_j with the form of expression '$-R_j$' also exists and resource R_j must exist in PN_jM_p and PN_jM_i for other candidate samples $M_p$in M, resource $R_j$in M_i can be obtained by extension of prior candidate function N_j.

Similarly, if a resource in the current candidate function to be extended meets the form of '$-R_1$', it should be judged whether this resource can be pruned according to

the weight of prior candidate function. Therefore, the following theorem can be used for pruning.

Lemma 3. Assuming that P is the bicluster with variant usage rate to be extended currently; M is the candidate function set of P and N is the prior candidate function set of P. If the form of expression is '$-R_j$' for any resource R_j in candidate function $M_i(M_i \in M)$ and there is a prior candidate function $N_j(N_j \in N)$ under which resource R_j with the form of expression '$-R_j$' also exists and resource R_j must exist in PN_jM_p and PN_jM_i for other candidate samples $M_p$in M, resource $R_j$in M_i can be obtained by extension of prior candidate function N_j.

Lemma 4. Assuming that P is the bicluster with variant usage rate to be extended currently; M is the candidate function set of P and N is the prior candidate function set of P. If the same prior candidate function $N_j(N_j \in N)$ exists for any resource R_j in candidate function $M_i(M_i \in M)$, making each resource R_j in candidate function M_i meet pruning conditions in Lemma 1 or 2 or 3, and $PN_jM_i.Resources$与$PM_i.Resources$are the same, candidate function M_i can be pruned.

It can be seen from theorem 4 that, the candidate function can only be pruned if all resources in the candidate function can be obtained by resource extension in the same prior candidate function; otherwise, this candidate function will be extended. If no successor or prior is its superset, the bicluster can be outputted. The mining process for *DeCluster* mining table 1 expressed matrix is shown in Fig.3. The specific description of *DeCluster* algorithm is as follows:

```
Algorithm 1: DeCluster algorithm
Input: number threshold: n, coherent threshold：α, low
usage rate threshold: β, variant usage rate threshold: r,
function-resource matrix: D
Output: all biclusters with maximal variant usage rate or
maximal low usage rate meeting the threshold
Initial value: sample weighted graph: G =Null, current
bicluster to be extended Q =Null, Si=Null and Sj=Null.
Algorithm description: DeCluster(n, α, β, r, D, Q, Si, Sj)
 (1) If G is null, scan data set D and make its weight
graph.Si is the first sample in the weighted graph;
 (2) For each sample Sj connected with sample Si
 (3)      If all resource linked lists in Sj satisfy prun-
ing conditions in Lemma 4, then
 (4)         Continue;
 (5)      Else
 (6)         For resource linked lists not satisfying prun-
ing conditions, Q.Sample= Q.Sample∪Sj；Q. Resource=
Q.Resource∩SiSj.Resource;
 (7)         DeCluster(n, α, β, r, D, Q, Si, Sj);
```

```
(8)    Endif
(9) Endfor
(10) If Q meets output conditions, then
(11)   Output Q
(12) Endif;
(13) S_i = S_i->next;
(14) Return
```

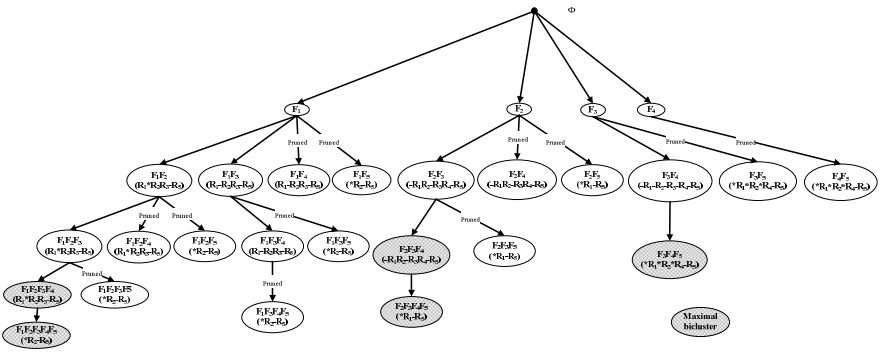


Fig. 2. The example mining procedure of DeCluster algorithm

4 Experimental Results

In this section, we will make an experimental comparison on the mining efficiency and result of the algorithm above and existing algorithms. To fully test the performance of the algorithm, we produce three data sets randomly, each of which contains 20 sampling sites and 1000 resources. Table 5 describes proportions of 0, 0.1, 0.2 and 0.8 in each row in each data set. In this section, a comparison will be made on the mining efficiency of *DeCluster* algorithm and *RAP* algorithm. To fully compare the extendibility of algorithms, we produce multiple groups of data sets with different numbers of resources and sampling sites in allusion to three data sets in table 5. The selection of resources and sampling sites are based on the order of resources and sampling sites in data set. The parameter of variant usage rate is 4 and that of low usage rate is 0.5.

Table 5. The proportion of each value in three data set

	0	0.1	0.2	0.8
D_1	0.2	0.2	0.2	0.4
D_2	0.2	0.3	0.3	0.2
D_3	0.4	0.2	0.2	0.2

Figs 3(a)-3(b) provide the comparison of performance period when the number of functions of two algorithms above is 10 and 20 respectively and the number of resources is 200, 400, 600, 800 and 1000 respectively and the parameter of relevancy under data set D_1 is 1. It can be seen from these figures that the mining time of both

algorithms increases progressively with the increase of number of resources in data set. Meanwhile, the mining efficiency of *DeCluster* algorithm is higher than that of RAP algorithm under each data size. Especially when the number of resources in data set is high, the mining efficiency of *DeCluster* algorithm is nearly 20 times higher than that of RAP algorithm. The reason is that RAP algorithm mines bicluster with the method of resource extension. With the increase of number of resources in data set, this algorithm needs more iterations to mine all biclusters meeting threshold conditions. However, *DeCluster* algorithm uses high-efficiency pruning strategies for mining and will produce more maximal biclusters especially when the number of resources in data set is high and data are dense. Therefore, *DeCluster* algorithm has a higher pruning efficiency. Figures 4(a)-4(b) provide the comparison of performance period under data sets with different resources of functions and resources when the parameter of relevancy of three algorithms above is 2 in data set D_1. Similar to the description in figs 3(a)-3(b), the mining efficiency of *DeCluster* algorithm is higher than that of RAP algorithm under each data size. When the number of resources in data set is low, the pruning efficiency of *DeCluster* algorithm is not significantly higher than that of RAP algorithm. However, with the increase of number of resources in data set, the pruning efficiency of *DeCluster* algorithm becomes significantly higher.

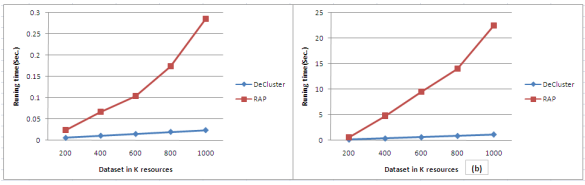


Fig. 3. The running time comparison between two algorithms under different number of resources and functions in D_1 when α=1: (a) 10 functions; (b) 20 functions

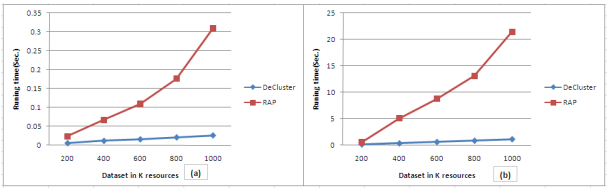


Fig. 4. The running time comparison between two algorithms under different number of resources and functions in D_1 when α=2: (a) 10 functions; (b) 20 functions

Figs. 5(a)-5(b) and figs. 6(a)-6(b) respectively provide the comparison of performance period of both algorithms above under data sets with different numbers of sampling sites and resources when their parameters of relevancy under data set $D_2$are respectively 1 and 2. It can be seen that, as proportions of 0.1 and 0.2 in data set D_2 increase compared to those in data set D_1, according to descriptions of the definition of variant usage rate and low usage rate, mining data set D_2will produce more biclusters than mining data set D_1 under the same parameter. Therefore, when the number of functions is 20, *RAP* algorithm cannot mine data sets with the number of resources

higher than 400 in limited memory space, but *DeCluster* algorithm can complete all mining processes within 10 seconds. Figs. 7(a)-7(b) and figs. 8(a)-8(b) respectively provide the comparison of performance period of both algorithms above under data sets with different numbers of sampling sites and resources when their parameters of relevancy under data set D_3 are respectively 1 and 2.

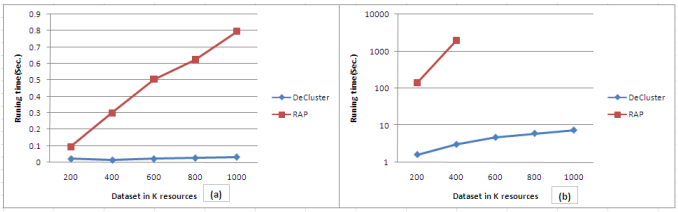


Fig. 5. The running time comparison between two algorithms under different number of resources and functions in D2 when α=1: (a) 10 functions; (b) 20 functions

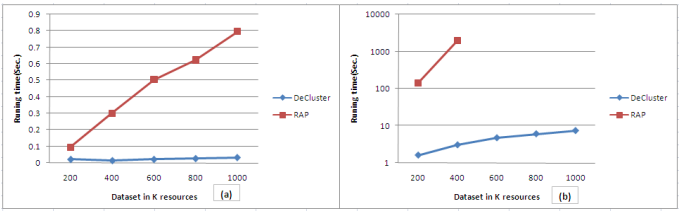


Fig. 6. The running time comparison between two algorithms under different number of resources and functions in D2 when α=2: (a) 10 functions; (b) 20 functions

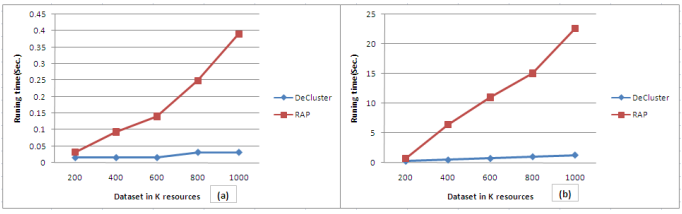


Fig. 7. The running time comparison between two algorithms under different number of resources and functions in D_3 when α=1: (a) 10 functions; (b) 20 functions

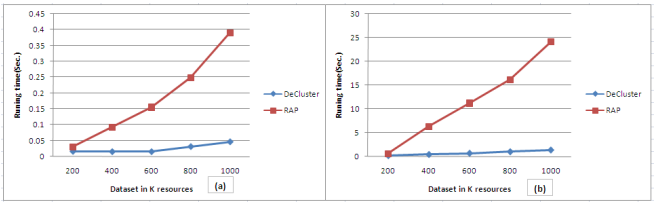


Fig. 8. The running time comparison between two algorithms under different number of resources and functions in D_3 when α=2: (a) 10 functions; (b) 20 functions

5 Conclusion

This paper proposed a new bicluster mining algorithm - *DeCluster* algorithm, which can effectively mine all biclusters with maximal variant usage rate and maximal low usage rate from the function-resource matrix of true value. First, this algorithm constructs a sample weighted graph which includes all resource collections between both samples that meet the definition of variant usage rate or low usage rate; then, all biclusters with maximal variant usage rate and low usage rate meeting the definition are mined with the mining method of using sample-growth and depth-first method in the constructed weighted graph. In order to improve the mining efficiency of the algorithm, *DeCluster* algorithm uses several pruning strategies to ensure the mining of maximal bicluster without candidate maintenance. Our next research direction is mining biclusters with variant usage rate and low usage rate in function-resource matrix measured in true environment.

References

1. Pecht, M., et al.: A prognostics and health management roadmap for information and electronics-rich systems. Microelectronics Reliability, 317–323 (2010)
2. Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology (ISMB 2000), pp. 93–103. ACM Press (2000)
3. Ben, et al.: Discovering local structure in gene expression data: the order-preserving submatrix problem. J. Comput. Biol. 10, 373–384 (2003)
4. Cheng, et al.: Bivisu: software tool for bicluster detection and visualization. Bioinformatics 23, 2342–2344 (2007)
5. Zhao, L., Zaki, M.J.: MicroCluster: An Efficient Deterministic Biclustering Algorithm for Microarray Data. IEEE Intelligent Systems, Special Issue on Data Mining for Bioinformatics 20(6), 40–49 (2005)
6. Pandey, G., Atluri, G., Steinbach, M., Myers, C.L., Kumar, V.: An association analysis approach to biclusting. In: Proc. ACM Conf. on Knowledge Discovery and Data Mining, pp. 677–686 (2009)
7. Torgeir, R.H., Astrid, L., Jan, K.: Learning rule-based models of biological process from gene expression time profiles using Gene Ontology. Bioinformatics 19, 1116–1123 (2002)
8. Wang, M., Shang, X., Zhang, S., Li, Z.: FDCluster: Mining frequent closed discriminative bicluster without candidate maintenance in multiple microarray data-sets. In: ICDM 2010 Workshop on Biological Data Mining and its Applications in Healthcare, pp. 779–786 (2010)
9. Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology (ISMB 2000), pp. 93–103. ACM Press (2000)
10. Wang, M., Shang, X., Miao, M., Li, Z., Liu, W.: FTCluster: Efficient Mining Fault-Tolerant Biclusters in Microarray Dataset. In: Proceedings of ICDM 2010 Workshop on Biological Data Mining and its Applications in Healthcare, pp. 1075–1082 (2011)
11. Wang, J., Han, J.: BIDE: Efficient Mining of Frequent Closed Sequences. In: Proceedings of the Data Engineering, pp. 79–90 (2004)

MEIT: Memory Efficient Itemset Tree for Targeted Association Rule Mining

Philippe Fournier-Viger[1], Espérance Mwamikazi[1], Ted Gueniche[1], and Usef Faghihi[2]

[1] Department of Computer Science, University of Moncton, Canada
[2] Department of Computer Science, Sull Ross State University, TX, USA
philippe.fournier-viger@umoncton.ca,
{eem7706,etg8697}@umoncton.ca, ufaghihi@sulross.edu

Abstract. The Itemset Tree is an efficient data structure for performing targeted queries for itemset mining and association rule mining. It is incrementally updatable by inserting new transactions and it provides efficient querying and updating algorithms. However, an important limitation of the IT structure, concerning scalability, is that it consumes a large amount of memory. In this paper, we address this limitation by proposing an improved data structure named MEIT (*M*emory *E*fficient *I*temset *T*ree). It offers an efficient node compression mechanism for reducing IT node size. It also performs on-the-fly node decompression for restoring compressed information when needed. An experimental study with datasets commonly used in the data mining literature representing various types of data shows that MEIT are up to 60 % smaller than IT (43% on average).

Keywords: frequent pattern mining, association rule mining, itemset mining, itemset tree, memory constraint, targeted queries.

1 Introduction

Association rule mining [1] is a fundamental data mining task with wide applications [2]. It consists of discovering associations in a transaction database. To mine association rules in a database, a user has to provide two thresholds, namely the minimum confidence and minimum support thresholds [1]. Several algorithms have been proposed to discover association rules such as Apriori, FPGrowth, HMine, Eclat and TopKRules [1, 2, 3, 4, 10, 11]. However, those algorithms are batch algorithms, i.e. if new data is added to the input database, users need to run the algorithms again to get updated results. This is inefficient when new data is regularly added to databases. To address this problem, incremental versions of batch algorithms were proposed [5, 6]. Nevertheless, these algorithms still suffer from an important limitation. That is, they are designed to discover (and update) all association rules in a database (meeting the user-defined thresholds mentioned above) rather than allowing the user to perform targeted queries. Targeted queries are useful for applications where the user wants to discover association rules involving a subset of the items contained in a database, instead of all items [7, 8]. To process targeted queries for association rule mining

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 95–106, 2013.

efficiently in the context of static or incremental databases, the *I*temset *T*ree (IT) data structure was proposed [7, 8]. The IT is a tree structure, which can be incrementally updated and efficiently queried. The IT structure allows performing a vast array of important targeted queries such as (1) calculating the frequency of a given set of items, (2) discovering all valid association rules given a set of items as antecedent and (3) finding all frequent itemsets subsuming a set of items and their support [7, 8]. The IT structure has various applications such as predicting missing items in shopping carts in real-time [9]. However, ITs are inefficient when it comes to memory efficiency. Thus, to use ITs for large and/or incremental databases, we need to improve their memory efficiency. Given this limitation, an important research question is: Could we design a more memory efficient structure for targeted association rule mining? In this paper, we answer this question positively by proposing an improved IT structure that we name the *M*emory *E*fficient *I*temset *Tree* (MEIT).

The contributions of this work are twofold. First, we propose the MEIT structure. It incorporates effective tree node compression and decompression mechanims to reduce the information stored in IT nodes and restore it when needed. Second, we perform an extensive experimental study on six datasets commonly used in the data mining literature to compare the MEIT and IT data structures. Results show that MEIT is up to 60 % smaller than an IT (43% on average).

The remainder of this paper is organized as follows. Section 2 reviews the problem of association rule mining and the definition of IT. Section 3 describes the MEIT. Section 4 presents the experimental study. Finally, Section 5 draws a conclusion and discusses future work.

2 Related Work

Association rule mining is a fundamental data mining problem [2]. It is stated as follows [1]. Let $I = \{a_1, a_2, \ldots a_n\}$ be a finite set of items. A transaction database is a set of transactions $T=\{t_1,t_2\ldots t_m\}$ where each *transaction* $t_j \subseteq I$ $(1 \leq j \leq m)$ represents a set of items purchased by a customer at a given time. An *itemset* is an unordered set of distinct items $X \subseteq I$. The *support count* of an itemset X is denoted as $sup(X)$ and is defined as the number of transactions that contain X. An *association rule* $X \rightarrow Y$ is a relationship between two itemsets X, Y such that $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The *support of a rule* $X \rightarrow Y$ is defined as $sup(X \rightarrow Y) = sup(X \cup Y) / |T|$. The *confidence of a rule* $X \rightarrow Y$ is defined as $conf(X \rightarrow Y) = sup(X \cup Y) / sup(X)$. The *problem of mining association rules* [1] is to find all association rules in a database having a support no less than a user-defined threshold *minsup* and a confidence no less than a user-defined threshold *minconf*. For instance, Figure 1 shows a transaction database (left) and the corresponding association rules (right) for *minsup* = 0.5 and *minconf* = 0.5. For example, rule $\{1\} \rightarrow \{4\}$ has a support of 0.5 because it appears in two transactions out of 6 (*t1* and *t4*). Furthermore, it has a confidence of 0.75 because {1, 4} appears in two transactions while {1} appears in 3 transactions. Mining associations is generally done in two steps [1]. Step 1 is to discover all frequent itemsets in the database (itemsets appearing in at least *minsup* × |T| transactions). Step 2 is to generate association rules using the frequent itemsets found in Step 1. For each frequent itemset X, pairs of

frequent itemsets P and Q = X – P are selected to generate rules of the form P→Q. For each such rule P→Q, if *sup*(P→Q) ≥ *minsup and conf*(P→Q) ≥ *minconf*, the rule is output. The most popular algorithms for association rule mining such as FPGrowth, HMine, Eclat and Apriori [1, 2, 3, 4, 10] are batch algorithms.

TID	Transactions
t1	{1, 4}
t2	{2, 5}
t3	{1, 2, 3}
t4	{1, 2, 4}
t5	{2,5}
t6	{2,4}

ID	Rule	Supp.	Conf.
r1	{2}→{5}	0.33	0.4
r2	{5}→{2}	0.33	1.0
r3	{1}→{4}	0.33	0.66
r4	{4}→{1}	0.33	1.0
r5	{2}→{4}	0.33	0.4
r6	{4}→{2}	0.33	0.66

Fig. 1. A transaction database (left) and some association rules f*ou*nd (right)

As an alternative to batch algorithms, the *Itemset-Tree* data structure was proposed. It is a structure designed for efficiently processing targeted queries on a transaction database [7, 8]. An IT is built by recursively inserting transactions from a transaction database, or any other sources, into the tree. It can be incrementally updated by inserting new transactions after the initial tree construction. An IT is formally defined as a tree where each IT node k stores (1) an itemset $i(k)$, (2) the support count $s(k)$ of the itemset and (3) pointers to children nodes when the node is not a leaf. The itemset associated to an IT node represents a transaction or the intersection of some transactions [7]. The root of an itemset tree is always the empty set ∅.

Figure 2 shows the algorithm for inserting a transaction in an IT. For instance, Figure 3 shows the six steps for the construction of an IT for the database depicted in the left part of Figure 1. In Step A, the transaction {1, 4} is inserted as a child of the root with a support of 1. In Step B, the transaction {2, 5} is inserted as a child of the root with a support of 1. In Step C, the transaction {1, 2, 3} is inserted into the tree. Because {1, 4} and {1, 2, 3} share the same leading item {1} according to the lexical ordering of items, a new node is created for the itemset {1} with a support of 2, such that {1, 2, 3} and {1, 4} are its children. In Step D, the transaction {1, 2, 4} is inserted into the tree. Given that {1, 2, 3} and {1, 2 4} share the same first leading items according to the lexical ordering, a node {1, 2} is created with a support of 2 with nodes {1, 2, 3} and {1, 2, 4} as its children. In Step E, the transaction {2, 5} is inserted into the tree. Since the transaction is already in the tree, its support count is incremented by 1 and no node is created. Finally, in Step F, the transaction {2, 4} is inserted. Since this transaction shares the itemset {2} with {2, 5}, a node {2} with a support of 3 is created with {2, 4} and {2, 5} as its children. Note that when the support of a node is increased in an IT, the support of all its ancestors is also increased. The expected cost of transaction insertion in an IT is ≈ O(1) [7]. For a proof that the transaction insertion algorithm is correct, the readers are referred to the paper proposing the IT structure [7].

An IT allows performing efficient queries for itemset mining and association rule mining such as (1) calculating the frequency of a set of items, (2) discovering all valid association rules containing a set of items as antecedent and (3) finding all frequent

itemsets subsuming a set of items and their frequency. Because of space limitation, we here only briefly explain the query-processing algorithm for counting the support of an itemset. The other query processing algorithms work in a similar way and the readers are referred to [7, 8] for more details. The pseudo-code of the query processing algorithm for support counting is shown in Figure 4. Consider the case of calculating the support of itemset {1, 2}. The algorithm starts from the root. Since the query itemset {1, 2} is not contained in and is smaller than the root itemset, the algorithm will visits the root's child nodes, which are {1, 2} and {2, 5}. Then given that the query itemset is equal to the itemset of the node {1, 2}, the support of 2 attached to the node {1, 2} will be kept and the subtree of that node will not be explored. The subtree {2, 5} will not be explored because the last item of {2, 5} is larger than the last item of {1, 2}. The algorithm will terminate and return 2 as the support count of {1, 2}. The expected cost of calculating the frequency of an itemset with this algorithm is ≈ O(n), where n is the number of distinct items in it. Improved querying algorithms for IT have recently been proposed [8]. Our proposal in this paper is compatible with both the old and the new querying algorithms.

INSERT(a transaction to be inserted s, an itemset-tree T)
1. $r = root(T)$; $s(r) := s(r) + 1$;
2. **IF** $i(s) = i(r)$ **THEN exit**;
3. Choose $T_s = subtree(r)$ such that $i(root(T_s)$ is comparable with s
4. **If** T_s does not exist **THEN**
5. create a new son x for r, $items(x) = s$ and $f(x) = 1$;
6. **ELSE IF** $i(root(T_s)) \subset s$ **THEN** call Construct(s, T_s)
7. **ELSE IF** $s \subset i(root(T_s)$ **THEN** create a new node x as a son of r
8. and a father of $root(T_s)$; $i(x) := s$; $s(x) := s(root(T_s))+1$;
9. **ELSE** create two nodes x and y, x as the father of $root(T_s)$,
10. such that .$i(x) = s \cap root(T_s)$, $s(x) = s(root(T_s))+1$,
11. and y as a son of x, such that $i(y) = s$, $s(y) = 1$.

Fig. 2. The algorithm for transaction insertion in an Itemset-tree

Lastly, note that IT should not be confused with *trie-based* structures used in pattern mining algorithms such as the FP-Tree structure used by FPGrowth [3]. In an FP-Tree, each node contains a single item and each branch represents a transaction. In an IT, each node represents a transaction or the intersection of some transactions [7, 8]. The trie-based structures and the IT are designed for a different purpose.

3 The Memory-Efficient Itemset-Tree

We now describe MEIT, our improved IT data structure for targeted association rule mining. MEIT is an enhanced form of IT that uses an efficient and effective node compression scheme to reduce the memory usage of IT. We have designed the MEIT based on three observations that are formalized by the following three properties.

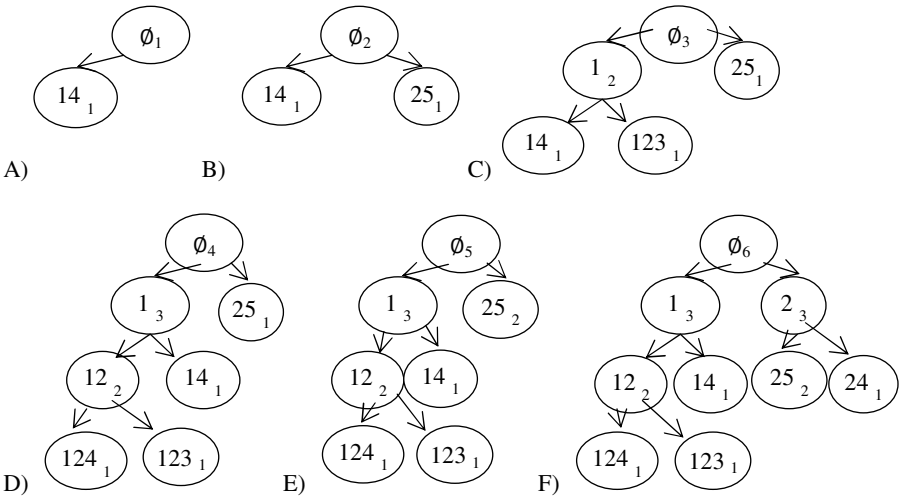


Fig. 3. An example of itemset-tree construction

COUNT(an itemset *s*, an itemset-tree *T*)
12. *r* = *root*(*T*);
13. **IF** *s* ⊆ *r* **THEN** *s*(*s*) := *s*(*s*)+ *s*(*r*);
14. **IF** *r* < *s* according to lexical order **AND** *last-item*(*r*) < *last-item*(*s*) **THEN**
15. **FOR EACH** subtree *T* of *r* **DO**
16. *s*(*s*) := *s*(*s*) + **COUNT**(*s, T*);
17. **RETURN** *s*(*s*);

Fig. 4. The algorithm for processing a support count query on an IT

Property 1. In an IT, transactions are inteserted by traversing branches in a top-to-bottom manner. **Rationale**. The insertion algorithm is given in Figure 3 and detailed in [7]. As it can be seen from the pseudo-code of this algorithm, the tree is always traversed in a top-to-bottom order to find the appropriate location for inserting a transaction (either by creating new node(s) or by raising the support of an existing node).

Property 2. Queries on an IT are always processed by traversing tree branches in a top-to-bottom manner. **Rationale**. In Figure 4, we have presented the pseudo-code of the algorithm for processing a support count query. As it can be seen from the pseudo code, the branches from the tree are traversed from top to bottom rather than from bottom to top. Other querying algorithms are described in [7, 8] and they also respect this property.

Property 3. Let *k* be an IT node and *parent*(*k*) be its parent. The relationship *i*(*parent*(*k*)) ⊂ *i*(*k*) holds between *k* and its parents. More generally, this property is transitive. Therefore, it can be said that for any ancestor *x* of *k*, *i*(*x*) ⊂ *i*(*k*).

Example 1. Consider the itemset {1, 2, 4} of the leftmost leaf node of Figure 3(F). The itemset of this node contains the itemset {1, 2} of its parent node. The itemset of this latter node contains the itemset {1} of its parent node. This latter contains the itemset ∅ of its parent.

Based on the aforementioned properties, we propose an efficient scheme for compressing node information in IT to improve its memory efficiency. It comprises two mechanisms, which are node compression and node decompression.

Definition 1. Node Compression. Consider a node k of an itemset tree having an uncompressed itemset $i(k)$ such that $i(k) \neq \emptyset$ (i.e. k is not the root). Suppose k has n ancestor nodes denoted as $ancestor_1(k)$, $ancestor_2(k)$, … $ancestor_n(k)$. Compressing node k consists of setting $i(k)$ to $i(k) \setminus \bigcup_{m=1}^{n} i(ancestor_m(k))$.

Example 2. For instance, Figure 5 shows the construction of an IT where the node compression scheme is applied on each new node during the IT construction, for the dataset of Figure 1 (left). During Step A and Step B, the transaction {1, 4} and {2, 5} are inserted into the IT with no compression because their parent is the empty set. In Step C, the transaction {1, 2, 3} is inserted. A new node {1} is created as a common ancestor of {1, 2, 3} and {1, 4}. The nodes {1, 2, 3} and {1, 4} are thus compressed respectively as {2, 3} and {4}. In Step D, the transaction {1, 2, 4} is inserted. A new node {2} is created as a common ancestor of {1, 2, 3} and {1, 2, 4} (note that {1, 2, 4} is represented as {4} in the tree because it is compressed). The nodes {1, 2, 3} and {1, 2, 4} are compressed respectively as {3} and {4}. In Step E, the transaction {2, 5} is inserted. Because this transaction already appears in the tree, the support of the corresponding node and its ancestors is increased. Finally, in Step F, the transaction {2, 4} is inserted. A new node {2} is created as a common ancestor of {2, 4} and {2, 5}. The nodes {2, 4} and {2, 5} are compressed respectively as {4} and {5}. By comparing the compressed trees of Figure 5 with the corresponding itemset tree of Figure 3(F), we can see that the total number of items stored in nodes is greatly reduced by compression. The tree of Figure 5 contains 8 items, while the tree of Figure 3(F) contains 16 items.

Having described the node compression scheme, we next describe how decompression is performed to restore the original information.

Definition 2. Node Decompression. Consider a node k of an itemset tree having a compressed itemset $i(k)$ such that $i(k) \neq \emptyset$ (i.e. k is not the root). Suppose that k has n ancestor nodes denoted as $ancestor_1(k)$, $ancestor_2(k)$, … $ancestor_n(k)$. Node decompression consists of calculating $i(k) \cup [\bigcup_{m=1}^{n} i(ancestor_m(k))]$ to obtain the uncompressed representation of $i(k)$.

Example 3. Consider the leftmost leaf node of Figure 5(F). It contains the itemset {4}, which is the compressed representation of the itemset {1, 2, 4} (cf. Figure 3(F)).

To restore the uncompressed representation, the union of the itemsets of the ancestor nodes with the itemset {4} is performed. The result is $\emptyset \cup \{1\} \cup \{2\} \cup \{4\} = \{1, 2, 4\}$. Note that the root can be ignored when performing the union of ancestor itemsets. This is because the root node always contains the empty set, and thus never changes the result of node decompression.

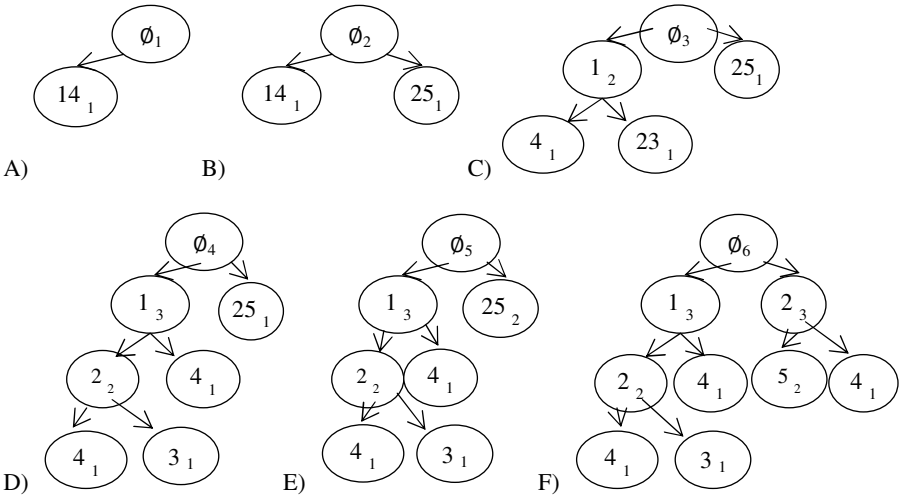


Fig. 5. An example of memory efficient itemset-tree construction

Definition 3. ***M*emory *E*fficient *I*temset *T*ree (MEIT)**. A MEIT is an IT where (1) node compression is applied during tree construction so that each node content is compressed and (2) where node decompression is performed on-the-fly during tree construction and query processing to restore node information when required.

We next show that node compression/decompression can be applied during tree construction and query processing, and that applying compression/decompression does not affect the result of queries.

Property 4. Node Compression Is Always Applicable during Transaction Insertion. To demonstrate that node compression can always be applied to compress new nodes that are created by the transaction insertion algorithm, we need to examine the conditions that have to be met to perform node compression. The first condition that has to be met to apply node compression to a given node is that the node has at least a parent node. The only node that does not have a parent in an IT is the root node and the root is never compressed by definition (cf. Definition 1). Second, to perform compression of a node *k*, it is necessary to access ancestor nodes' itemsets to perform their union. This condition is always met during tree construction because branches are always traversed from top to bottom rather than from bottom to top (Property 1). Therefore, the itemsets of ancestor nodes can be collected while traversing each branch so that the information required for compression is always available to the

transaction insertion algorithm. Therefore, any node that is visited by the transaction insertion algorithm can be compressed when it is created.

Property 5. Node Decompression Is Always Applicable for Transaction Insertion. To demonstrate that node decompression can always be applied by the transaction insertion algorithm, we use a similar reasoning as for Property 4. As mentioned, the transaction insertion algorithm traverses branches from top to bottom starting from the root (Property 1). This traversal order makes it possible to collect the itemsets of nodes visited while traversing a branch so that ancestor nodes' itemsets are always available for performing decompression of a node. Therefore, any node that is visited by the transaction insertion algorithm can be decompressed.

Property 6. Node Decompression Is Always Applicable during Query Processing. For query processing, only decompression is used to answer queries. It must be noted that by definition queries are not allowed to modify an IT (transaction insertion is not viewed as a query in IT terminology [7]). Similar to transaction insertion, queries are always processed by traversing IT branches from top to bottom rather than from bottom to top (Property 2), as it can be seen for example in the pseudo-code of Figure 4. Because of this traversal order, the information required for decompressing each node can be collected as the branches are traversed by keeping itemsets from the same branch into memory. Therefore, any node that is visited by one of the query processing algorithm can be decompressed.

Property 7. Performing Node Compression/Decompression Does Not Affect Query Results. Compressing an itemset and decompressing it does not result in a loss of information, by the definition of compression and decompression. Itemsets are compressed during transaction insertion and are decompressed on the fly when needed during transaction insertion and query processing. Because of this, the process of compression/decompression is completely transparent to the operations of the transaction insertion and query processing. Thus, it does not affect query results.

Implementing Node Compression Efficiently. In the following, we explain why the complexity of node compression is linear. When a new node k is inserted into an MEIT, the cost of compression consists of performing the union of the itemset to be inserted m with the itemsets of the n ancestors of k, to calculate $i(k) = m / \bigcup_{p=1}^{n} i(ancestor_p(k))$. To perform the union of the ancestor itemsets efficiently, one can notice that when a node is inserted, all ancestor nodes already in the tree have been compressed. Given that compressed nodes do not share items with their ancestors, the union of the ancestor itemsets can be performed simply by a concatenation in linear time rather than by an expensive union operation. Now let's consider how to perform the set subtraction of the itemset $\bigcup_{p=1}^{n} i(ancestor_p(k))$ from m. To perform this operation efficiently, itemsets in the tree should always be sorted in lexicographical order. If itemsets are in lexicographical order (or any other total order), set subtraction can be performed efficiently by the means of a two-way comparison, which requires scanning m and the itemset

$\bigcup_{p=1}^{n} i(ancestor_p(k))$ at most one time. Thus the complexity of set subtraction is O(m+k) and the complexity of concatenation is linear.

Implementing Node Decompression Efficiently. The complexity of node decompression is also linear. Let k be a node having n ancestors $ancestor_1(k)$, $ancestor_2(k)$, … $ancestor_n(k)$ when the tree is traversed from top to bottom from the root to k. To restore the itemset compressed in node k, it is necessary to perform the union of all itemsets stored in its ancestor nodes $ancestor_1(k)$, $ancestor_2(k)$, … $ancestor_n(k)$ with $i(k)$, as previously mentioned. Performing the union of several sets can be costly if implemented naïvely. To implement the union efficiently, we suggest using the following strategy. All items stored in each tree node should be sorted according to the lexical ordering (or any other total order). Then, while recursively traversing the tree from top to bottom to reach k, the union can be efficiently performed by simply concatening the itemsets $i(ancestor_1(k))$, $i(ancestor_2(k))$, … $i(ancestor_n(k))$ with $i(k)$, in that order. Thus the cost of node decompression is O(m), where m is the cardinality of the decompressed itemset. In practice, the cost of decompression is even smaller because intermediate concatenation results can be kept into memory when traversing a branch from top to bottom. For instance, consider a node k and its parent $parent(k)$. The concatenation of $i(k)$ and $parent(i(k))$ needs only to be performed once for all descendant nodes of k. This implementation strategy can greatly improve efficiency.

4 Experimental Study

We have implemented MEIT and IT in Java. The IT and MEIT source codes as well as all the datasets used in the experiments can be downloaded from http://goo.gl/hDtdt as part of the open-source SPMF data mining software. The following experiments are performed on a computer with a Core i5 processor running Windows 7 and 1 GB of free RAM. All memory measurements were performed using the core Java API. Experiments were carried on real-life and synthetic datasets commonly used in the association rule mining literature, namely *Accidents*, *C73D10K*, *Chess*, *Connect*, *Mushrooms*, *Pumsb* and *Retail*. Table 1 summarizes their characteristics.

Experiment 1. Memory Usage Comparison. We first compared the size of IT and MEIT for all datasets. To assess the efficiency of node compression, we measured the total number of items (including duplicates) stored in IT and MEIT nodes for each dataset. Results are shown in Table 1. As it can be seen in the third column of Table 1, the compression of nodes achieved by MEIT varies from 26.4 % to 88.7 % with an average of 58.7 %. The largest compression is achieved for dense datasets (e.g. *Mushrooms*), while less compression is achieved for sparse datasets (e.g. *Retail*). This is because transactions in dense datasets share more items. Thus, the intersection of transactions is larger than in sparse datasets. Thus, each IT node generally contains more items than for a sparse dataset. This gives more potential for compression in the MEIT.

Because MEIT only compress itemsets inside nodes and nodes contain other information such as pointers to child nodes, it is also important to compare the total

memory usage of MEIT and IT. To do that, we measured the total memory usage of IT and MEIT. Results are shown in Table 2. The third column shows that a compression of 13 % to 60 % is achieved, with an average of 43 %. We can conclude that the compression of itemset information has a considerable impact on the total memory usage.

Table 1. Datasets' Characteristics

Dataset	Transaction count	Distinct items count	Average transaction size
Accidents	340,183	468	22
C73D10K	10,000	1,592	73
Chess	3,196	75	37
Connect	67,557	129	43
Mushrooms	8,416	128	23
Pumsb	49,046	7,116	74
Retail	88,162	16,470	172

Table 2. Memory Usage of IT and MEIT (total items stored)

Dataset	IT size (items)	MEIT size (items)	Size reduction (%)
Accidents	16057697	5262555	67.2%
C73D10K	882196	508878	42.3%
Chess	196579	39550	79.9%
Connect	4446791	1092208	75.4%
Mushrooms	308976	35003	88.7%
Pumsb	4046316	2773440	31.5%
Retail	938568	690889	26.4%

Table 3. Total Memory Usage of IT and MEIT (MB)

Dataset	IT size (MB)	MEIT size (MB)	Size reduction (%)
Accidents	84.1	43.0	49%
C73D10K	4.0	2.6	36%
Chess	1.0	0.4	60%
Connect	21.8	9.0	59%
Mushrooms	1.7	0.7	60%
Pumsb	18.3	13.5	26%
Retail	7.3	6.4	13%

Experiment 2. Compression Overhead. We next compared the construction time of MEIT and IT for each dataset to assess the overhead in terms of execution time incurred by node compression and decompression. Results are shown in Table 3. As it can be seen, the overhead during tree construction is generally more or less the same for each dataset, averaging 45 %. We expected such an overhead because additional operations have to be performed to compress and decompress node information on-the-fly. We also compared the query processing time of MEIT and IT for 10,000 random queries for each dataset to assess the overhead of on-the-fly decompression in terms of execution time for query processing. Results are shown in Table 4. As it

can be seen, the overhead for query processing is generally more or less the same for each dataset, averaging 44 %. Again, we expected such an overhead because of extra operations performed for on-the-fly node decompression in MEIT. For real applications, we believe that this overhead is an excellent trade-off given that it allows building itemset-trees that can contains up to twice more information into memory (cf. Experiment 1). Moreover, as it can be seen in this experiment, the overhead in terms of execution time is predictable. It is more or less the same for each dataset no matter the size of the dataset or the type of data stored. In future work, we will assess the possibility of using caching algorithms to store the uncompressed form of frequently accessed nodes to reduce the overhead for popular queries.

Table 4. Tree construction time and 10K query processing time for IT/MEIT

Dataset	Tree construction time (s)		Time for processing 10K queries (s)	
	IT	MEIT	IT	MEIT
Accidents	3.71	5.82	880.4	1696.8
C73D10K	0.25	0.38	23.8	39.9
Chess	0.22	0.30	2.6	5.0
Connect	0.56	0.82	153.5	231.1
Mushrooms	0.21	0.23	1.0	2.1
Pumsb	0.63	0.95	134.2	202.1
Retail	4.33	7.75	84.8	193.7

5 Conclusion

An efficient data structure for performing targeted queries for itemset mining and association rule mining is the Itemset Tree. However, a major drawback of the IT structure is that it consumes a large amount of memory. In this paper, we addressed this drawback by proposing an improved data structure named the Memory Efficient Itemset Tree. During transaction insertion, it employs an effective node compression mechanism for reducing the size of tree nodes. Moreover, during transaction insertion or query processing, it relies on an on-the-fly node decompression mechanism for restoring node content.

Our experimental study with several datasets that are commonly used in the data mining literature shows that MEIT are up to 60 % smaller than IT, with an average of 43 %. In terms of execution time, results show that the overhead for on-the-fly decompression is predictable. The amount of overhead is more or less the same for each dataset no matter the amount of data or the type of data stored. We believe that the overhead cost is an excellent trade-off between execution time and memory given that it allows building itemset-trees that can store up to twice the amount of information for the same amount of memory. For future work, we will explore other possibilities for compressing IT such as exploiting the links between nodes. We also plan to develop a caching mechanism, which would store the decompressed form of the most frequently visited nodes to improve efficiency for popular queries. We also plan to develop new querying algorithms for targeted top-k association rule mining [11, 12].

Source code of IT and MEIT as well as all the datasets used in the experiments can be downloaded from http://goo.gl/hDtdt as part of the open-source SPMF data mining software.

Acknowledgment. This work has been financed by an NSERC Discovery grant from the Government of Canada.

References

1. Agrawal, R., Imielminski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: Proc. ACM Intern. Conf. on Management of Data, pp. 207–216. ACM Press (1993)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
3. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation. Data Mining and Knowledge Discover 8, 53–87 (2004)
4. Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., Yang, D.: H-Mine: Fast and space-preserving frequent pattern mining in large databases. IIE Transactions 39(6), 593–605 (2007)
5. Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of discovered association rules in large databases: An incremental updating technique. In: Proceedings of the Twelfth International Conference on Data Engineering, pp. 106–114. IEEE Press (1996)
6. Ezeife, C.I., Su, Y.: Mining incremental association rules with generalized FP-tree. In: Cohen, R., Spencer, B. (eds.) Canadian AI 2002. LNCS (LNAI), vol. 2338, pp. 147–160. Springer, Heidelberg (2002)
7. Kubat, M., Hafez, A., Raghavan, V.V., Lekkala, J.R., Chen, W.K.: Itemset trees for targeted association querying. IEEE Transactions on Knowledge and Data Engineering 15(6), 1522–1534 (2003)
8. Lavergne, J., Benton, R., Raghavan, V.V.: Min-Max itemset trees for dense and categorical datasets. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 51–60. Springer, Heidelberg (2012)
9. Wickramaratna, K., Kubat, M., Premaratne, K.: Predicting missing items in shopping carts. IEEE Transactions on Knowledge and Data Engineering 21(7), 985–998 (2009)
10. Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: Proc. of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 326–335. ACM Press (2003)
11. Fournier-Viger, P., Wu, C.-W., Tseng, V.S.: Mining Top-K Association Rules. In: Kosseim, L., Inkpen, D. (eds.) Canadian AI 2012. LNCS, vol. 7310, pp. 61–73. Springer, Heidelberg (2012)
12. Fournier-Viger, P., Tseng, V.S.: Mining Top-K Non-Redundant Association Rules. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 31–40. Springer, Heidelberg (2012)

Mining Frequent Patterns in Print Logs with Semantically Alternative Labels

Xin Li[1], Lei Zhang[1], Enhong Chen[1], Yu Zong[2], and Guandong Xu[3]

[1] University of Science and Technology of China
[2] West Anhui University
[3] University of Technology, Sydney
{leexin,stone,cheneh}@ustc.edu.cn, nick.zongy@gmail.com,
guandong.xu@uts.edu.au

Abstract. It is common today for users to print the informative information from webpages due to the popularity of printers and internet. Thus, many web printing tools such as *Smart Print* and *PrintUI* are developed for online printing. In order to improve the users' printing experience, the interaction data between users and these tools are collected to form a so-called print log data, where each record is the set of urls selected for printing by a user within a certain period of time. Apparently, mining frequent patterns from these print log data can capture user intentions for other applications, such as printing recommendation and behavior targeting. However, mining frequent patterns by directly using url as item representation in print log data faces two challenges: data sparsity and pattern interpretability. To tackle these challenges, we attempt to leverage delicious api (a social bookmarking web service) as an external thesaurus to expand the semantics of each url by selecting tags associated with the domain of each url. In this setting, the frequent pattern mining is employed on the tag representation of each url rather than the url or domain representation. With the enhancement of semantically alternative tag representation, the semantics of url is substantially improved, thus yielding the useful frequent patterns. To this end, in this paper we propose a novel pattern mining problem, namely mining frequent patterns with semantically alternative labels, and propose an efficient algorithm named **PaSAL** (Frequent **Pa**tterns with **S**emantically **A**lternative **L**abels Mining Algorithm) for this problem. Specifically, we propose a new constraint named conflict matrix to purify the redundant patterns to achieve a high efficiency. Finally, we evaluate the proposed algorithm on a real print log data.

Keywords: print log data, frequent pattern mining, delicious.com, PaSAL.

1 Introduction

With wide applications of internet and office automation tools, printing informative information from the web pages is becoming popular today. There are many web printing tools such as *Smart Print*[1] and *PrintUI*[2] are developed for online printing. With the use of these tools, many interaction data by users are collected under certain consent. These

[1] www.smartprint.com/
[2] www.printui.com/

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 107–119, 2013.

Table 1. User and Printed URLs

User	Printed URLs
$user_1$	http://maps.google.com/maps?saddr=27400+Old+Trilby+Roa
	http://www.groupon.com/deals?city=houston
	http://www.booking.com/hotel/us/the-huston.en.html
$user_2$	.....
	http://maps.google.com/maps?saddr=2800+League+City+Parkwa
.....	.....

Table 2. URL domains and Their Tag Representations

Domain	Tag Representations
maps.google.com	maps, google, travel, map, reference, search, directions, tools, clear-lake, brian
www.groupon.com	shopping, coupons, deals, social, discount, coupon, business, crowd-sourcing, marketing, travel
www.booking.com	travel, hotel, hotels, booking, accommodation, search, viajes, online, hoteles, turismo

interaction data is also called Print log data, where each record is the set of urls selected for printing by a user within a certain period of time. Table 1 gives an example of print log data. Compared with the search log data [13], the print log data is a new kind of user interaction data and is attracting researchers' attention.

Since the webpages printed by users may reveal their interests, mining frequent patterns from these print log data becomes one of important methods capturing users' interests and has benefiting other applications such as printing recommendation and behavior targeting. However, we find that using the url directly as the item representation for frequent pattern mining faces two challenges in real print log data, namely, data sparsity and pattern interpretability. For example, as shown in Table 1, $user_1$ prints the map of "Old Trilby Roa" while $user_2$ prints the map of "League City Parkwa", both of them print their destination maps on google maps and take along with them. However, the two printed urls from the two users are completely different and cannot be recognized as the same item when running any kinds of the frequent pattern mining algorithms unless we manually label. In addition, patterns using urls as items representation has a poor interpretability.

To that end, we attempt to enrich the semantics of each url by utilizing the semantically alternative tags associated with the domain of each url to form the representation of url. More precisely, when processing a url, we extract and input the domain of this url to *delicous.com api* [3] and obtain the top-10 returned tags as the representation. Table 2 shows three domains in Table 1 and their corresponding returned tags. By replacing each domain of the url with a set of tags and removing redundant tags in a transaction, the print log of url transactions is transformed to be a tag transaction dataset. Mining frequent patterns on this new tag transaction data is more practical, since these tags labeled by humans are abundant and meaningful.

If we run frequent pattern mining algorithms on tag transaction data without considering any extra constraint, we could get massive patterns and most of them are redundant and meaningless. For example, if the domain *maps.google.com* in Table 2 is frequent, then any combination of tags (e.g., {*maps,google*}, {*maps, google, travel*})

[3] Delicious.com api: `https://delicious.com/developers`

referred to the domain is also frequent, resulting in a larger number of trivial frequent patterns (2^{10}). Therefore we introduce a conflict matrix to reflect these restrict conditions, which will be defined in Section 3. To this end, we propose a novel frequent pattern mining technique for print log data, named **PaSAL** (Frequent **Pa**tterns with **S**emantic **A**lternative **L**abels Mining Algorithm) highlighted by enhancing the url representation via semantically alternative tags and incorporating the constraint of conflict matrix for further pruning the lattice during the mining process. In summary, we make following contributions.

• We define a novel frequent pattern mining problem for print log data with semantically alternative labels. In order to solve the problem of data sparsity and pattern interpretability, we use the returned tags from *delicious.com api* as the representations for urls.

• We devise an efficient algorithm called PaSAL for the above problem. We define a new constraint named conflict matrix for pruning meaningless patterns. PaSAL can exploit this constraint to further reduce the search space during the mining process, thus achieving a high efficiency.

• We conduct several experiments on a real print log data to evaluate our proposed algorithm. The experimental results show that our method outperforms the baseline and achieves a better pattern interpretability.

The paper is organized as follows. In Section 2, we describe the related work. In Section 3, a formal description about the problem will be given. We give the basic algorithm and our algorithm PaSAL in Section 4. We evaluate the effectiveness and efficiency of PaSAL in Section 5 and conclude the paper in Section 6.

2 Related Work

Frequent Pattern Mining Algorithm. Mining association rule was first proposed by Agrawal et al. in [1]. Since then, there are many algorithms have been developed to mine frequent patterns, such as Apriori [2] and FP-growth [5]. Instead of using horizontal data format, Zaki et al. proposed to use vertical data format in [14]. Of course, researchers try to mine maximal frequent patterns [3] and closed frequent patterns [7] in order to avoid mining massive patterns. Our work is based on these works yet using a bitmap representations of database.

Semantic Information Extension. Besides inventing new algorithms to mine patterns, Han et al. [4] and Srikant et al. [11] proposed structured model of the items in order to mine patterns at a different level, which can be regarded as introducing external information to help mining patterns. Our work is different from their work in two aspects. First, the previous works have multi-level structures while we only have one. Second, items in the structure are owner-member relationship, however tags in our restrict conditions are semantically relevant to each other. In addition, Mei et al. [6] interpret the frequent patterns by giving semantic annotations through natural language processing, which is a post process while we represent each url before mining. Information can either be generated by machine inferring from the history log or artificial rules. In our paper, we use delicious tags to expand the semantic information of url, which is newly demonstrated.

Constraint Based Pruning. Another related work is about constraint on pattern mining process, which is mentioned in works of Pei et al. [8] [9] and Raedt et al. [10]. In this paper, we propose a new constraint named *conflict matrix* for further pruning the lattice. In the following, we will show that the conflict matrix constraint is a anti-monotone one, which can be deeply exploited in the mining process to achieve high efficiency. Besides, [12] proposed to mine dominant and frequent patterns based on the data from the Web printing tool Smart Print. However, the dateset from [12] is very different from our. Specifically, in [12] each record of the dataset is the selected clips (contents) for a certain Webpage while that in this paper is a set of printing URLs from one user.

3 Problem Statement

First, we will give some preliminaries about frequent pattern mining. Let $\mathcal{T}$ be the complete set of transactions and $\mathcal{I}$ be the complete set of distinct items. Any non-empty set of items is called an *itemset* (or *pattern*) . The support of a pattern *P* is the percentage of transactions that contain *P*. Frequent pattern mining is to find frequent patterns whose *support* is above the user specified threshold.

Note that the primitive transaction database is a URL database, where each transaction is the set of URLs selected by a user. If we use the domain of each URL as the representation, the URL database can be transformed as a domain database (denoted as $\mathcal{D}$). As is mentioned above, when you put a domain into delicious.com api, the server will return top-10 tags as representations referred to the domain you submitted. By replacing domain by a set of tags $\{t_1, t_2, ..., t_k\}$ (k is equal to 10 in most instances), the domain database is transformed as a tag database. Note that if a domain is frequent in the domain database, then all the subsets of tags in this domain is also frequent. In order to reducing the number of patterns like this, we propose to use Conflict Matrix CM to purify the frequent pattern mining results. Formally,

Definition 1 (Conflict Matrix). *Conflict Matrix CM is a matrix with $|\mathcal{I}| \times |\mathcal{D}|$ dimensions. Here $|\mathcal{T}|$ denotes the number of tags while $|\mathcal{D}|$ counts the number of domains. The value in the matrix is defined as follows:*

$$CM_{i,j} = \begin{cases} 1 & t_i \in T(d_j) \\ 0 & t_i \notin T(d_j) \end{cases} \tag{1}$$

where t_i is a tag and $T(d_j)$ is a set of tags associated with domain d_j.

Definition 2 (Conflict Matrix Constraint). *Given a pattern P and a conflict matrix CM, P is a valid pattern if there is no subset of P that is included in any one of transactions in CM.*

In order to achieve high efficiency, we can store CM with bitmap representation.

Example. As shown in Table 1 and 2, we sample some few tags for the 10 returned tags to make an example due to the limit of the page. Let $\mathcal{D} = \{maps.google.com, www.groupon.com, www.booking.com\}$ and $\mathcal{I}$ = $\{maps, travel, search, hotel, coupon\}$, so we get CM, displayed in Equation 2.

$$CM = \begin{array}{c} \\ maps \\ travel \\ search \\ hotel \\ coupon \end{array} \begin{array}{c} \begin{array}{ccc} google & groupon & booking \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array} \tag{2}$$

Consider the database shown in Table 3(a) and let user specified minimum support be 0.3 (that is, 2 transactions at least). Then the patterns with minimum support (frequent patterns) are shown in Table 3(b). Note that travel, search co-occur in both domain google and booking, thus it should be removed from our candidate basket. The same thing happened on maps, search and hotel, search. Finally, we get two patterns marked with ✓in Table 3(b) and those are what we want to get. Somehow we can say tags in a transaction of RC are mutual exclusion.

Table 3. A Sampled Example

(a) Sampled Database $\mathcal{T}^{trans}$

Transaction Id	Tags
100	maps, hotel, search
200	maps, hotel, coupon
300	travel, search, hotel, maps
400	coupon, search
500	search, travel, coupon
600	travel, search

(b) Frequent Patterns

Pattern	Support
✓maps, hotel	2
✓coupon, search	2
maps, search	2
travel, search	3
hotel, search	2

(c) Example

No.	Items
t_1	b c
t_2	a b
t_3	a b c
t_4	a b d
t_5	a b c d

Problem Statement. Given a set of transactions $\mathcal{T}$, conflict matrix constraint CM and a user-specified minimum support (min_sup), the problem of mining frequent patterns with semantically alternative labels is to find all frequent patterns that do not violate conflict matrix constraints. Note that if CM is null, the problem degenerates into a normal frequent pattern mining problem.

The problem we defined here has many applications. One immediate application is widely used in supermarket, that is how to put goods on shelves reasonably. Goods displaying has many strict limits, like our restrict matrix. For example, when people go to supermarket buying some food, they may bring raw beef, vegetables and fried chicken together home. However, raw food and cooked food are absolutely forbidden to put together. Hence, though we mine many frequent patterns from shopping list transaction, we need to think carefully before we act to do cross-selling. A useful way is to take restrict matrix into consideration.

4 Algorithms

In the rest of the section, we first show the naive approach for the proposed novel problem, and then present our efficient algorithm PaSAL.

4.1 Algorithm Basic

The basic algorithm is designed below.

1. First, find all the frequent patterns by any of the frequent pattern mining algorithms like “Apriori” or “FP-growth”.

Algorithm 1. BASIC

input : the tag transaction data $\mathcal{T}$, lexicographic ordering $\mathcal{I}$, the conflict matrix CM
output: restrict condition constrained patterns $\mathcal{P}$

```
1  Root.head ← empty;
2  Root.tail ← I;
3  DFS(Root);
4  DFS(Node) begin
5      for tag ∈ Node.tail do
6          p_tmp ← Node.head ∪ tag;
7          if p_tmp.frequency > min_sup × |T| then
8              Node_tmp.head ← p_tmp;
9              remove tag from Node_tmp.tail ← Node.tail;
10             Children ← Node_tmp;
11             P_candidate ← Node_tmp.head;
12     for node ∈ Children do
13         DFS(node);
14 for pattern ∈ P_candiate do
15     if pattern violate CM then
16         remove pattern from P_candidate;
17 P ← P_candidate;
```

2. Then, check each frequent pattern whether violates the restrict matrix constraint and obtain the ultimate result.

Algorithm 1 gives an overview of the whole process, using the notation in Section 3. In this algorithm, we set the activate node be the root of the lattice, with its head be empty and tail be the lexicographic ordering of all tags in $\mathcal{T}$. Then we iteratively visit each tag in its tail. By adding the tag to its head, we count the cardinality of its binary set after bitwise-AND operation and filter out those node whose cardinality is below the user-specified threshold (min_sup $\times|\mathcal{I}|$). Patterns (Nodes) without been filtered will be added to $\mathcal{P}_{candidate}$ and $Children$, where $\mathcal{P}_{candidate}$ is used for post process and nodes in $Children$ is waiting for recursive procedure. After filtering the nodes without minimum support, we actually prune the lattice by abandoning all the descendants, which is under the basic assumption that supersets of infrequent itemsets are also infrequent.

At last, every pattern is checked in $\mathcal{P}_{candidate}$ whether it violates restrict matrix constraint. Specifically, we do bitwise-AND on all the tag vectors from CM in a pattern to see if the cardinality of the binary set is zero. If so, it means that tags have no overlap on any restrict conditions, thus it will be kept or it will be removed. Finally, we get $\mathcal{P}$ as a result.

4.2 Algorithm PaSAL

Instead of using Basic algorithm, we propose a new algorithm called PaSAL. In this algorithm, we do pruning during the traversal rather than post process in basic one. This is carried out by exploiting the anti-monotonicity of conflict matrix and the bitmap data structure for storing conflict matrix.

Anti-monotonicity. In order to do pruning, we aim to estimate the restrict condition or the conflict matrix of all nodes in lattice *L*. The basic idea is described as follows.

First, we propose a label to calculate the extent of the conflict to a certain pattern, denote as $\sigma(P)$. $\sigma(P)$ can be calculated in this way. Supposing that $P = \{t_1, t_2, ..., t_m\}$ and each tag t_i in pattern P is a conflict vector in conflict matrix CM, so $\sigma(P)$ is the number of domains that conflict take places, which can be formalized as:

$$\sigma(P) = \sum_{1 \le i,j \le m} card(t_i \,\&\, t_j) \tag{3}$$

In Equation 3, & is bitwise-AND operation and $card(*)$ means the number of 1's in a bitwise vector $*$.

Second, we need to find the properties of label $\sigma(P)$. Since $\sigma(P)$ is defined above, we can easily infer that $\sigma(P)$ is not decreasing as the pattern grows as shown in Property 1.

Property 1. For any pattern *P* and its extension t_e, the label $\sigma(*)$ satisfies that

$$\sigma(P) \le \sigma(P + t_e) \tag{4}$$

Proof. Let's consider the definition of $\sigma(P)$ in Equation3. Still $P = \{t_1, t_2, ..., t_m\}$

$$\begin{aligned}
&\sigma(P + e) \\
&= \textstyle\sum_{1 \le i,j \le m+1} card(t_i \,\&\, t_j) \\
&= \textstyle\sum_{1 \le i,j \le m} card(t_i \,\&\, t_j) + \sum_{1 \le i \le m} card(t_i \,\&\, t_m + 1) \\
&= \textstyle\sum_{1 \le i,j \le m} card(t_i \,\&\, t_j) + \sum_{1 \le i \le m} card(t_i \,\&\, t_e) \\
&\ge \textstyle\sum_{1 \le i,j \le m} card(t_i \,\&\, t_j) \\
&= \sigma(P)
\end{aligned}$$

Note that t_e here is same to t_{m+1}, and $\sum_{1 \le i \le m} card(t_i \,\&\, t_e)$ is definitely no less than zero. □

Last, similar to metrics support, we assign each pattern that do not violate any constraint a value called "traversability", which is define as follows:

$$traversability(P) = 1/\sigma(P) \tag{5}$$

Thus traversability satisfies the property 2

Property 2. Property "traversability" in Equation 5 is anti-monotonicity.

Proof. Since $\sigma(P)$ is nomotonicity, the reciprocal of $\sigma(P)$ is anti-monotonicity. □

Hence, Property 2 can be used for pruning the lattice. The key point of the algorithm is to set a flag in each node of the lattice in order to distinguish if the node violates the conflict matrix constraint by adding the tag from the tail to head. The process is shown in Algorithm 2.

Bitmap Operations. Since each tag could find its own binary vector from CM, we can conduct the calculation easily by bitwise-AND and bitwise-OR. As described in Equation 1, each row tag vector in CM indicates which domain contains the referred tag. The position is set to binary one when the domain contains the tag else zero. More intuitional feeling can be got in Equation 2.

By doing **bitwise-AND**, we first judge the conflict between two tags. In Equation 2, also known as an example, the vector of tag *maps* equals to $\{1, 0, 0\}$ and vector of tag *travel* is $\{1, 1, 1\}$, when operating bitwise-AND, we get $\{1, 0, 0\}$, which means that

Algorithm 2. PaSAL

input : the tag transaction data $\mathcal{T}$, lexicographic ordering $\mathcal{I}$, the conflict matrix CM
output: restrict condition constrained patterns $\mathcal{P}$

```
1  Root.head ← empty;
2  Root.tail ← I;
3  Root.flag ← (0)_2;
4  DFS(Root);
5  DFS(Node) begin
6      for tag ∈ Node.tail do
7          p_tmp ← Node.head ∪ tag;
8          flag_tmp ← (Node.flag & CM_tag)_2;
9          if p_tmp.frequency > min_sup × |T| and flag_tmp == 0 then
10             Node_tmp.head ← p_tmp;
11             remove tag fromNode_tmp.tail ← Node.tail;
12             Node_tmp.flag ← (Node.flag | CM_tag)_2;
13             Children ← Node_tmp;
14             P_candidate ← Node_tmp.head;
15     for node ∈ Children do
16         DFS(node);
17 P ← P_candidate;
```

they violate the restrict condition on domain "google". The judging process is corresponding to line 8-9 in Algorithm 2. If we get vector full of binary zero, we say two tags are compatible.

Then we operate **bitwise-OR** to aggregate two vectors to gain more restricted vector and wait for another tag to be added, which is shown in line 12. We assign the vector to the flag of descendant node whose tags comes from his father's head and one tag from father's tail. Of course, minimum support should be taken into consideration at the same time when doing the judge.

Next, we add descendants satisfying those two rules into $\mathcal{P}_{candidate}$ the same as in Algorithm 1. There is no need for us to do extra post process to purify the final result $\mathcal{P}_{candidate}$, that is $\mathcal{P} = \mathcal{P}_{candidate}$.

4.3 Discussion

Bitmap Representation. Bitmap representation is chosen for the transaction data. In doing this way, each transaction corresponds to one bit. If item **i** appears in transaction **T**, the T^{th} position of bitmap for item i is set to binary one, otherwise, the position is set to zero. This character is naturally fit for the print log data.

Here we use bitmap representation twice in our method. First, tag transaction data is transformed to bitmap, thus we can calculate the support easily. Supposing that we get two tags t_1 and t_2, each with a bitmap vector, denoted as bitmap(t_1) and bitmap(t_2). So we get bitmap($t_1 \bigcup t_2$) = bitmap(t_1) & bitmap(t_2), where & is a bitwise-AND. The same thing happened when an itemset (pattern) meets a tag, which is common when we add an element in a node's tail to its head. Thus bitmap($head \bigcup tag$) = bitmap($head$) & bitmap(tag). In the following, we only need to calculate the cardinality of $head \bigcup tag$,

where cardinality is number of 1's in a bitmap vector. Second, the restrict condition RC is also shown in bitmap representation, i.e., conflict matrix CM. By doing this way, it's much convenient for us to judge whether it violates the previously stipulated condition when adding a element (tag) from a node's tail to its head. Let bitmap of a tag in conflict matrix be bitmap(CM_{tag}) and a bitmap of a node's flag be bitmap($node.flag$), then do bitwise-AND between bitmap(CM_{tag}) and bitmap($node.flag$). If the cardinality of the bitmap result equals to zero, there is no conflict, otherwise, they must conflict at some transaction (domain). If no conflict happens, we do bitwise-OR between bitmap(CM_{tag}) and bitmap($node.flag$) and then assign the bitmap value to node's descendant.

Pruning Lattice. In Figure 1 and Figure 2, an example of lattice is shown under lexicographic order. Each node with a bubble in right top corner denotes or an itemset. The number in the bubble is the frequency of the node according to Table 3(c). With the basic method, we search 12 nodes in total while the number of searched nodes decrease to 8 by using our method. By comparing the two figures, we can easily draw a conclusion that our method reduces the search space on lattice. This works well because we introduce another constraint during the traversal.

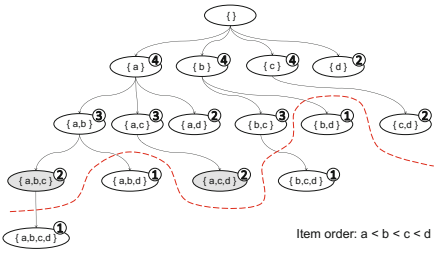


Fig. 1. Post Process on Lattice

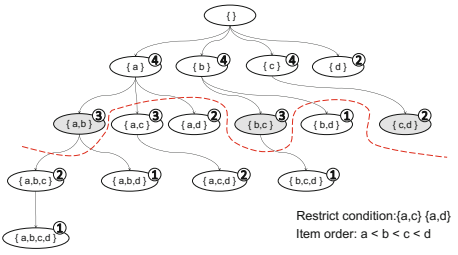


Fig. 2. In Process on Lattice

5 Experiment Evaluation

In this section we evaluate both the effectiveness and efficiency of our method. Our experiment was conducted on a real-world print log dataset. The characteristics of the data set is summarized in Table 4. All experiments were performed on a personal computer with a AMD Athlon X2 240 2.81GHz CPU and 4G of memory running the windows 7 operating system.

Table 4. Characteristics of the Print Log

Database Name	Number of Records	Number of Transactions
URLs	107031	23212
Domains	59416	23211
Tags	328752	16041

The three transaction database are generated from the same print log data. Database **Domains** and **Tags** are originated from database **URLs**. By removing the specific information in a url after slash, we get a domain. For instance, given a user printed url in Table 1, we get domain in Table 2 by using the method just mentioned. Then we use delicious.com api returned tag sets to expand each domain's semantic information, which is well shown in Table 2 and discussed in Section 1. According to our experiment, delicious.com tags cover more than 60% of the database **Domains**. In order to get fair treatment, we test our method on the database with the tag representations and ignore the rest part, thus the transaction number of the three database are equal.

5.1 Evaluation on Effectiveness

In order to prove the effect of our method in solving the problem of data sparsity and pattern interpretability, experiment can be divided into two parts: (1) the number of patterns by expanding the semantic information and (2) verifying pattern interpretability through case study. The experiment was conducted on three database of the same transaction number.

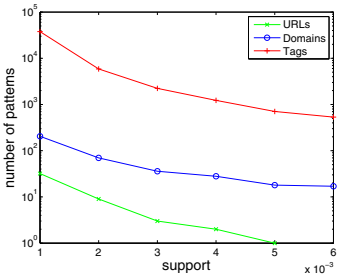


Fig. 3. Patterns mined from Three Database

Number of Patterns. In this subsection, we compare the number of frequent patterns on the three datasets with the different minimal frequent threshold min_sup. Any of the algorithms for mining frequent patterns can be chosen here and we implement a simple frequent pattern mining algorithm in [3]. Figure 3 shows the number of patterns mined from the three database in relevance to the minimum support thresholds. Value on y-axis is taken the logarithm. We set the threshold from 0.1% up to 0.6%. The result clearly show that after expanding the semantic information, we gain more patterns than the baselines.

Case Study. The frequent patterns mined on **URLs** dataset are all frequent 1-itemset, like '$\{maps.google.com/maps?hl = en\&tab = wl\}$ due to the data sparsity. When applying on database **Domains**, we get several patterns with frequency equal to 2, such as $\{maps.google.com, www.mapquest.com\}$. From the domain pattern, we can roughly infer that user may want some map service. For database **Tags**, we frequent pattern $\{maps, shopping, travel\}$. Now, besides knowing that people wants map service, we can tentatively say that the user is preparing for a tour since *shopping* exists here in

Table 5. Patterns Mined From Three Transaction Database

Database	Patterns
URLs	http://www.facebook.com/
	http://www.foodnetwork.com/food/cda/recipe-print/
	https://www.shopping.hp.com/webapp/shopping/
Domains	en.wikipedia.org, www.ehow.com
	www.amazon.com, www.ehow.com
	maps.google.com, www.bing.com
Tags	food, health, shopping
	facebook, google, networking, social
	google, maps, travel

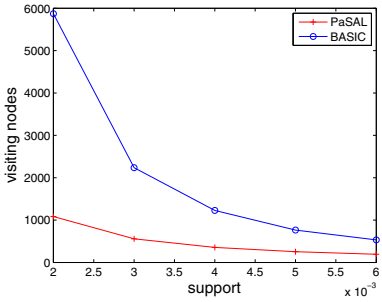


Fig. 4. Visiting Nodes

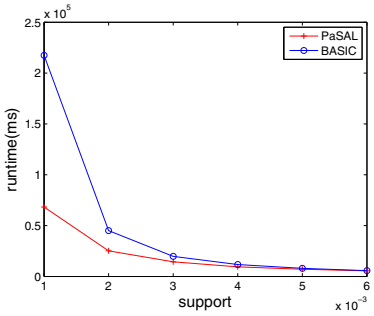


Fig. 5. Running Time

the set. By using tags to expand url's semantic information, we can not only gain more information explainable but also to understand a user's interest at fine-grained. More cases can be found in Table 5.

5.2 Evaluation on Efficiency

Since we use delicious tags as representations, we need to purify our mined result because the replacement is under the assumption that tags in a delicious returned set referred to a domain are semantically alternative or relevant to each other. In order to evaluate the efficiency of our proposed algorithm PaSAL, we choose the Brute-Force method as the baseline. This experiment was conducted only on the database **Tags**.

Here we choose the running time and pruning effect as the measures. Figure 5 shows that our algorithm run faster over 3 times than baseline when min_sup=0.0001. In order to show the effect of pruning, we also show the number of visiting nodes in Figure 4. We can see that by embedding conflict matrix constrait in the traversal we do more pruning on the lattice, thus leading to high efficiency.

The running time of two methods varies greatly when the minimum support is low. The reason is that there exists huge search space when setting min_sup low and this is what our algorithm good at. PaSAL can perform better pruning effect when meeting with large lattice, otherwise, the baseline running time is approaching our algorithm when min_sup increases. This is because that lots of time is spent on retrieving and calculating the bitmap structure in PaSAL.

6 Conclusion

In this paper, we defined a novel pattern mining problem, namely mining frequent patterns with semantically alternative labels. This problem is motivated by enriching the semantics of each url in print log data to solve the problem of data sparsity and pattern interpretability. Specifically, we attempt to utilize the semantically alternative tags returned from *delicious.com api* as the representation of each url. Then, we propose an efficient algorithm named PaSAL for this novel problem. Specifically, we propose a *conflict matrix* constraint to purify the redundant patterns and this constraint is then deeply exploited in the mining process to achieve high efficieny. Finally, we show the effectiveness and efficiency of the proposed algorithm on a real print log data. It is worth mentioning that the proposed algorithm PaSAL can be applied to many fields, such as super market cross-selling checking.

Acknowledgements. The authors Xin Li, Lei Zhang and Enhong Chen were supported by grants from Natural Science Foundation of China (Grant No. 61073110), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20113402110024), and National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant No. 2012BAH17B03). The author Yu Zong are supported by grants from the Nature Science Foundation of Anhui Education Department of China under Grant No. KJ2012A273 and KJ 2012A274; and the Nature Science Research of Anhui under Grant No. 1208085MF95. Enhong Chen gratefully acknowledges the support of Huawei Technologies Co., Ltd. (Grant No. YBCB2012086)

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB (1994)
3. Burdick, D., Calimlim, M., Gehrke, J.: Mafia: a maximal frequent itemset algorithm for transactional databases. In: International Conference of Data Engineering (2001)
4. Han, J., Fu, Y.: Discovery of multiple-level association rules from large databases. In: Proceedings of the 21th International Conference on Very Large Data Bases (1995)
5. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: SIGMOD (2000)
6. Mei, Q., Xin, D., Cheng, H., Han, J., Zhai, C.: Semantic annotation of frequent patterns. ACM Transactions on Knowledge Discovery from Data, TKDD (2007)
7. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
8. Pei, J., Han, J.: Can we push more constraints into frequent pattern mining? In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000)
9. Pei, J., Han, J.: Constrained frequent pattern mining: a pattern-growth view. ACM SIGKDD Explorations Newsletter, 31–39 (2002)

10. Raedt, L.D., Zimmermann, A.: Constraint-based pattern set mining. In: SIAM International Conference on Data Mining (2007)
11. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Proceedings of the 21st International Conference on Very Large Data Bases (1995)
12. Tang, L., Zhang, L., Luo, P., Wang, M.: Incorporating occupancy into frequent pattern mining for high quality pattern recommendation. In: CIKM (2012)
13. Wang, X., Zhai, C.: Learn from web search logs to organize search results. In: SIGIR, pp. 87–94 (2007)
14. Zaki, M.: Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12, 372–390 (2000)

Minimising K-Dominating Set in Arbitrary Network Graphs

Guangyuan Wang[1], Hua Wang[1], Xiaohui Tao[1], Ji Zhang[1], and Jinhua Zhang[2]

[1] Department of Maths and Computing
University of Southern Queensland, QLD, 4350, Australia
[2] School of Applied Mathematics, Beijing Normal University, Zhuhai, 519085, China
{guangyuan.wang,hua.wang,xiaohui.tao,ji.zhang}@usq.edu.au,
zhangjinhua471@sohu.com

Abstract. A self-stabilizing algorithm, after transient faults hit the system and place it in some arbitrary global state, recovers in finite time without external (e.g., human) intervention. A k-dominating set in a distributed system is a set of processors such that each processor outside the set has at least k neighbors in the set. In the past, a few self-stabilizing algorithms for minimal k-dominating set (MKDS) have been obtained. However, the presented self-stabilizing algorithms for MKDS work for either trees or a minimal 2-dominating set. Recently a self-stabilizing algorithm for MKDS in arbitrary graphs under a central daemon has been investigated. But so far, there is no algorithm for the MKDS problem in arbitrary graphs that works under a distributed daemon. In this paper, we propose a self-stabilizing algorithm for finding a MKDS under a distributed daemon model when operating in any general network graph. We further verify the correctness of the proposed algorithm (Algorithm MKDS) and prove that the worst case convergence time of the algorithm from any arbitrary initial state is $O(n^2)$ steps where n is the number of nodes in the network.

Keywords: Self-stabilizing algorithm, Minimal k-dominating set, Distributed daemon model, Arbitrary network, Convergence.

1 Introduction

Recently, social networks have received dramatic interest in research and development. Aiming at delivering better experience to customers and making business transformation along with Web 2.0, many information systems have adopted social networks [26,22,20]. Social networks, however, have also introduced to the research community many new challenges, for example, how to protect customers' privacy in such an electronic social environment [24,15,20]. Graph theory has been considered a working solution to tackle these challenges. The classical graph problem, such as the k-dominating set problem has many new practical applications.

For example, assume there is a need to build some fire stations in Toowoomba city such that its six areas can receive help from at least k times services from

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 120–132, 2013.

its neighboring areas. Assume there is also a fixed cost for building a fire station in each area. The problem is determining the number of the fire stations such that the total building cost is minimum among the participating areas under the k times services condition. This problem is equivalent to the problem of finding a minimum weighted k-dominating set among these areas. The following is the formal definition of k-dominating set.

1.1 Minimal k-Dominating Set Problem

Let $G = (V, E)$ be a connected simple undirected graph in which each node $i \in V$ represents an area and each edge $(i, j) \in E$ represents the bidirectional link connecting nodes i and j. Let k be an arbitrary positive integer. A subset D of V is a k-dominating set in G if each node not in D has at least k neighbors in D. A k-dominating set D in G is *minimal* if any proper subset of D is not a k-dominating set in G. The so-called *minimal k-dominating set (MKDS) problem* is to find a minimal k-dominating set (MKDS) in G. A k-dominating set for $k = 1$, i.e., a 1-dominating set, is just an ordinary dominating set.

For example, without loss of generality, considering the case $k = 2$, 2-dominating set in Toowoomba city as remarked early, which has 6 areas (a_1, a_2, ..., a_6) and the neighborhood relations are showed in the Fig. 1 . We can select the areas $\{a_3,\ a_4,\ a_5\}$ is a minimal 2-dominating set.

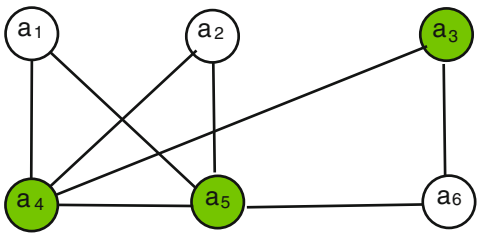


Fig. 1. A minimal 2-dominating set example

1.2 Self-stabilizing Algorithm

Self-stabilization is an optimistic fault tolerance approach for distributed systems. It was introduced by Dijkstra in [2,3]. According to his work, a distributed system is self-stabilizing if it can start at any possible global configuration and regain consistency in a finite number of steps by itself without any external intervention (e.g., human) and remains in a consistent state [4]. Recently, some self-stabilizing algorithms for dominating sets, independent sets, colorings, and matchings in graphs have been developed [11,8,1,10,12,19].

A fundamental idea of self-stabilizing algorithms is that the distributed system may be started from an arbitrary global state. After finite time the system

reaches a correct global state, called a *legitimate* or *stable* state. An algorithm is *self-stabilizing* if, when the system executes the algorithm, the following two properties hold: *convergence* and *closure*. That is,

(i) for any initial illegitimate state it reaches a legitimate state after a finite number of node moves (*convergence*), and
(ii) for any legitimate state and for any move allowed by that state, the next state is a legitimate state (*closure*).

In a self-stabilizing algorithm, each process maintains its local variables, and can make decisions based on the knowledge of its neighbors' states. A process changes its local state by making a *move* (a change of local state). The algorithm is a set of rules of the form "*if condition part (or guard)*" then "*action part*". The condition part (or guard) is a Boolean function over the states of the process and its neighbors; the action part is an assignment of values to some of the process's shared registers. A process i becomes *privileged* if it's condition is true. When a process becomes privileged, it may execute the corresponding move.

Various execution models have been suggested for developing self-stabilizing algorithms. These models are encapsulated within the notion of a daemon (or scheduler). The *distributed daemon* activates the processors by repeatedly selecting a set of processors and activating them simultaneously to execute a computation step. Each processor executes the next computation step as defined by its state just prior to this activation. Once every processor in the set has finished reading, all the processors write a new state (change state). Only then does the scheduler choose a new set of processors to be activated. Note that no non-activated processor changes its state. The *central daemon* is a special case of the distributed daemon in which the set of activated processors consists of exactly one processor. Thus if a system is self-stabilizing under the distributed daemon model, then it is self-stabilizing under the central daemon model. The converse, however, is not true (the total dominating set algorithm in [7] and the 2-dominating set algorithm in [13] are self-stabilizing under the central daemon models, but not under the distributed daemon models).

The rest of this paper is organized as follows. Section 2 presents our motivation and contribution. Section 3 presents a self-stabilization algorithm and an illustration for finding a minimal k-dominating set (MKDS). Section 4 proves the convergence and the time complexity of the proposed algorithm. Section 5 discusses the related work and algorithm comparison. Section 6 concludes the paper and discusses the future work.

2 Motivation and Contribution

Due to the publication of Dijkstra's pioneering paper, some graph problems have been solved by self-stabilizing algorithms in the literature, such as the self-stabilizing algorithms for independent sets and dominating sets in graphs [5,8,11,1,10]. Among these problems, Dominating Set and related problems are considered to be of central importance in combinatorial optimization and have

been the object of much research. Due to the NP-complete of domination problems [6], researchers have developed some self-stabilizing algorithms for finding minimal dominating sets [5,23,1,10].

Hedetniemi et al. [11] presented two *uniform* algorithms (a distributed algorithm is said to be *uniform* if all of the individual processes run the same code) for the dominating set (DS) and the minimal dominating set (MDS) problems. Lan et al. [18] presented a linear-time algorithm for the k-domination problem for graphs in which each block is a clique, a cycle or a complete bipartite graph. Kamei and Kakugawa [16] presented two self-stabilizing algorithms for the minimal k-dominating set (MKDS) problem in a tree. Huang et al. presented two self-stabilizing algorithms to find a minimal 2-dominating set (M2DS) in an arbitrary graph. Wang et.al developed a self-stabilizing algorithm for finding a MKDS in an arbitrary graph which works under a central daemon in 2012 [23].

However, there is no algorithm for the MKDS problem in arbitrary graphs that works under a distributed daemon. The reason for that is mainly due to the higher complexity of the execution of the algorithm under the distributed daemon model. The proposed algorithms for the MKDS work either for trees (Kamei and Kakugawa [16]) or under a central daemon (Wang et al. [23]).

In this paper, we extend Huang et al.'s work and consider the extension problem of minimal 2-dominating set (M2DS) just mentioned in [13,14]. We will solve that extension problem for general k (i.e., for k being an arbitrary positive integer) in general networks. We firstly develop a new self-stabilization algorithm for finding a minimal k-dominating set (MKDS) in a general network that works under a distributed daemon and analyze the correctness and time complexity of the proposed algorithm, in which the time complexity of their algorithm is not been discussed in [14]. We believe the following to be our contributions in this paper.

1. We present a self-stabilizing algorithm for finding a minimal k-dominating set (MKDS) under a distributed daemon in an arbitrary connected simple undirected graph.
2. We further verify the correctness of the proposed algorithm and prove that the worst case convergence time of the algorithm from any arbitrary initial state is $O(n^2)$ steps where n is the number of nodes in the network graph.

3 Self-stabilizing K-Dominating Set Algorithm

In this section, our self-stabilizing algorithm for solving the minimal k-dominating set problem will be presented.

3.1 Formal Definition of the Problem

The distributed system in consideration has a general underlying topology, and can be modeled by a connected simple undirected graph $G = (V, E)$, with each node $i \in V$ representing a processor in the system and each edge $(i, j) \in E$

representing the bidirectional link connecting processors i and j. It is assumed that the number of all processors in G is denoted by n. Assume now that for each processor $i \in V$, the set $N(i)$ represents its *open neighborhood*, denotes the set of processors to which i is adjacent. $d(i)$ represents the number of neighbors of processor i, or its degree ($d(i) = |N(i)|$). It is assumed that

(1) each processor i in the system has a unique identity,
(2) each processor i maintains two shared registers, d_i and p_i,
(3) $N(i)$ denotes the set of all open neighbors of i and $L(i) = \{j \in N(i) | j < i\}$,
(4) the value of d_i is taken from $\{0, 1\}$,
(5) $D(i) = \{j \in N(i) | d_j = 1\}$, and $|D(i)|$ is the cardinality of $D(i)$, and
(6) the value of p_i is always $\emptyset$, $\{i\}$ or $D(i)$.

A Boolean variable d_i indicates membership in the set D that we are trying to construct. The value $d_i = 1$ indicates that $i \in D$, while the value $d_i = 0$ indicates that $i \notin D$.

The concept of a minimal k-dominating set (MKDS) problem gives rise to the following problem:

Minimal K-Dominating Set

INSTANCE: A connected simple undirected graph $G = (V, E)$ and an arbitrary positive integer k.

QUESTION: How to find a minimal k-dominating set (MKDS) $D \subseteq V$ such that the D is a k-dominating set of the graph G and minimal, i.e., to construct $D = \{i \in V | d_i = 1\}$ is a MKDS.

3.2 Proposed Algorithm

The Algorithm MKDS is shown below which consists of five rules. Assume k being an arbitrary positive integer. It should also be reiterated that the computational model assumed in the system is the distributed daemon model.

Algorithm MKDS

k is an arbitrary positive integer. For each node i

$R1$: $d_i = 0 \wedge |D(i)| < k \wedge p_i = \{i\} \wedge \forall j \in \{m \in L(i) | d_m = 0\},\ p_j \neq \{j\} \rightarrow d_i := 1$

$R2$ $d_i = 1 \wedge |D(i)| \geq k \wedge \forall j \in N(i) - D(i),\ p_j = \emptyset \rightarrow d_i := 0$

$R3$: $d_i = 0 \wedge |D(i)| < k \wedge p_i \neq \{i\} \rightarrow p_i = \{i\}$

$R4$: $d_i = 0 \wedge |D(i)| = k \wedge p_i \neq D(i) \rightarrow p_i = D(i)$

$R5$: $d_i = 0 \wedge |D(i)| > k \wedge p_i \neq \emptyset \rightarrow p_i = \emptyset$.

$R1$ tries to ensure that a node $i \notin D$ which has less than k neighbors in D should enter D, if its pointer points to itself and its neighbors which have IDs smaller than its are all dominated at least k times. $R2$ says that a node i

is dominated at least k times by D should leave D if as far as it can tell all of its neighbors not in D are dominated at least k times. That is, the node i is redundant for its neighbors not in D since they are dominated at least k times by $D-\{i\}$. $R3$, $R4$ and $R5$ mean that the nodes not in D should reset their pointers according to the numbers of their neighbors in D. An example to illustrate the execution of Algorithm MKDS is shown in Fig. 2.

The example in Fig. 2 is to illustrate the execution of Algorithm MKDS. Without loss of generality, we consider the case $k = 2$ and use a minimal 2-dominating set example to illustrate the execution of Algorithm MKDS. Note that in each configuration, the shaded nodes represent privileged nodes.

In the first subgraph of Fig. 2, we set $d_1 = d_3 = 1$, other nodes' d-values are 0, that means $D = \{1,\ 3\}$. We further set $p_1 = p_5 = \{1\}$, $p_3 = p_6 = \{3\}$, $p_2 = \{2\}$, $p_4 = \{1,\ 3\}$, just as the arrows point in the first subgraph. According to the *Rules* of Algorithm MKDS, after a serial of moves and resets, the system reaches a legitimate state. As the last subgraph of Fig. 2 shows, which is a legitimate configuration, we can see a minimal 2-dominating set $D = \{1,\ 2,\ 3,\ 6\}$ can be identified. Note that the 2-dominating set $D = \{1,\ 2,\ 3,\ 6\}$ is minimal not minimum. The set $D = \{3,\ 4,\ 5\}$ or $D = \{4,\ 5,\ 6\}$ is a minimum 2-dominating set.

4 The Stabilization Time of Algorithm MKDS

The legitimate configurations are defined to be all those configurations in which no node in the system is privileged. The following theorem clarifies that in any legitimate configuration, a minimal k-dominating set can be identified (*closure*).

Theorem 1. *If Algorithm MKDS stabilizes,, then the set* $D = \{i \in V | d_i = 1\}$ *is a minimal* k*-dominating set.*

Proof. It is obvious that the system is in a legitimate configuration if and only if no node in the system is privileged.

(1) Suppose D is not a k-dominating set. Then there exists a node $i \in V - D$ such that i has at most $k-1$ neighbors in D, i.e., $d_i = 0$ and $|D(i)| < k$.

Claim. For any node i in the system when it reaches a legitimate configuration, if $d_i = 0$ and $|D(i)| < k$, then there exists a node $j \in L(i)$ (thus $j < i$) such that $d_j = 0$ and $|D(j)| < k$.

Proof of Claim. Since $d_i = 0$, $|D(i)| < k$, and i is not privileged by $R3$, we have $p_i = \{i\}$. Then since i is not privileged by $R1$, i.e., $\forall j \in \{m \in L(i) | d_m = 0\}, p_j \neq \{j\}$ cannot hold. Hence there exists a node $j_0 \in L(i)$ such that $d_{j_0} = 0$ and $p_{j_0} = \{j_0\}$. If $|D(j_0)| = k$, then since $d_{j_0} = 0$ and $p_{j_0} = \{j_0\} \neq D(j_0)$, j_0 is privileged by $R4$, which causes a contradiction. If $|D(j_0))| > k$, then since $d_{j_0} = 0$ and $p_{j_0} = \{j_0\} \neq \emptyset$, j_0 is privileged by $R5$, which causes a contradiction. Hence $|D(j_0))| < k$ and the claim is proved. □

By applying the above claim to node i, we get a node $i_1 \in L(i)$ such that $d_{i_1} = 0$ and $|D(i_1)| < k$. Then, by applying the claim to node i_1, we get a node

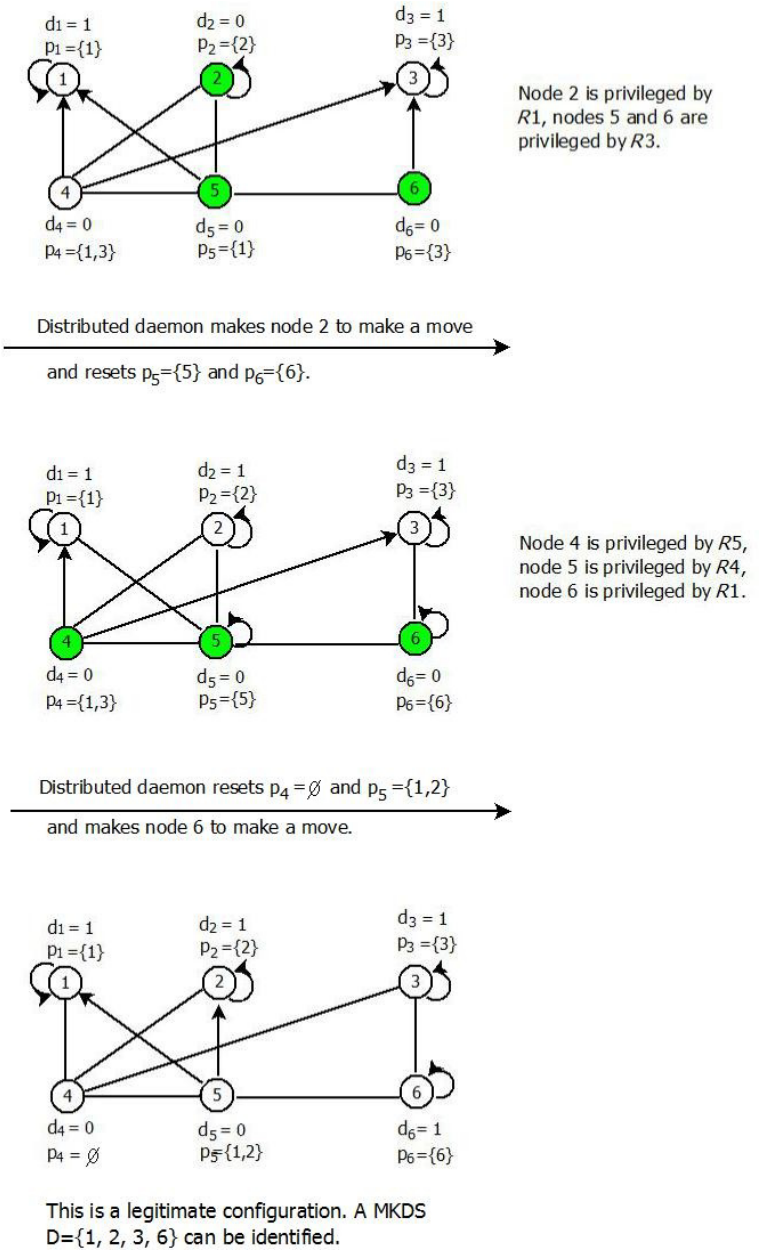


Fig. 2. A minimal 2-dominating set example to illustrate the execution of Algorithm MKDS

$i_2 \in L(i_1)$ such that $d_{i_2} = 0$ and $|D(i_2)| < k$. In this way, we eventually get infinitely many nodes $i_1, i_2, i_3, \ldots$ such that $i > i_1 > i_2 > i_3 > \cdots$. However, this causes a contradiction because the system has only a finite number of nodes. Therefore, D must be a k-dominating set.

(2) Suppose D is not a minimal k-dominating set. Then there exists a node $i \in D$ such that $D - \{i\}$ is a k-dominating set. Since $i \notin D - \{i\}$ and $D - \{i\}$ is a k-dominating set, i has at least k neighbors in $D - \{i\}$ and thus $|D(i)| \geq k$. If $N(i) - D(i) = \emptyset$, then, since $d_i = 1$ and $|D(i)| \geq k$, i is privileged by $R2$, which causes a contradiction. Hence $N(i) - D(i) \neq \emptyset$. Let j be an arbitrary node in $N(i) - D(i)$. Since $j \notin D - \{i\}$ ($d_j = 0$) and $D - \{i\}$ is a k-dominating set, j has at least k neighbors in $D - \{i\}$. Thus j has at least $k + 1$ neighbors in D, i.e., $|D(j)| > k$. Since $d_j = 0$, $|D(j)| > k$ and j cannot be privileged by $R5$, we have $p_j = \emptyset$. Hence the condition $[\ \forall j \in N(i) - D(i), p_j = \emptyset\ ]$ holds. Since $d_i = 1$, $|D(i)| \geq k$ and $[\ \forall j \in N(i) - D(i), p_j = \emptyset\ ]$ hold, node i is privileged by $R2$, which causes a contradiction. Hence D is a minimal k-dominating set. ■

In the following, we show the convergence of Algorithm MKDS. We will first provide two Lemmas. Based on these two Lemmas, we will prove the convergence time of Algorithm MKDS.

Lemma 1. *If d_i changes from 0 to 1, then d_i will not change again.*

Proof: Since d_i changes from 0 to 1 by $R1$ at time t, so the condition $d_i = 0 \wedge |D(i)| < k \wedge p_i = \{i\} \wedge \forall j \in \{m \in L(i) | d_m = 0\},\ p_j \neq \{j\}$ holds. By $R1$, only the node with larger ID than $i's$ is able to enter the set D. Suppose at the time t', the node i leaves the set D by $R2$, thus the condition $d_i = 1 \wedge |D(i)| \geq k \wedge \forall j \in N(i) - D(i),\ p_j = \emptyset$ holds. So there exists a node $j \in N(i) - D(i)$ with lager ID than $i's$ entering the set D. Then the node j satisfies $R1$, we can get $p_j = \{j\}$, but if the node i leaves the set D it satisfies the *R2*, and we have $p_j = \emptyset$, which causes a contradiction. ■

Lemma 2. *A node can make at most two membership moves.*

Proof: If a nodes first membership move is by $R1$, by Lemma 1, it will not make a membership move again. If its first membership move is R2, then any next membership move must be by $R1$, after which, it cannot make another membership move. ■

Now we will prove Algorithm MKDS always stabilizes (*convergence*).

Theorem 2. *Algorithm MKDS produces a minimal k-dominating set (MKDS) and stabilizes in $O(n^2)$ steps.*

Proof: In light of Theorem 1 we need only show stabilization (*convergence*). By Lemma 2, each node will change its d-value at most twice. Therefore, there can be at most $2n$ changes of d-values on all nodes in all the time. If there is no change in d-value of any node in a time-step, then the time-step involves only changes in p-values. The change in a p-value is determined only by d-values and

its neighbors in the set D according to the $R3$, $R4$ or $R5$. Consider the processor $i \in V$ with the largest degree Δ, a guarded command the p_i may execute is $R3$, $R4$ or $R5$. By definition of the Algorithm MKDS, after p_i executes a guarded command once, it is no longer privileged until $|D(i)|$ changes by an execution of at least one of the neighbors. Thus, the p_i execute $R3$, $R4$ or $R5$ at most 2Δ times in an infinite computation. So, the upper bound of the execution time is $2(\Delta+1)n$ time-steps. Considering the graph G is a connected simple undirected graph, the upper bound of the Δ is $(n-1)$, therefore, Algorithm MKDS produces a MKDS and the stabilization time of Algorithm MKDS is $O(n^2)$ steps. ■

5 Related Work and Algorithm Comparison

In this section, we collect and discuss the existing self-stabilizing algorithms for dominating sets respectively. We also compare our algorithm with theirs based on their basic ideas. The algorithms presented in this section are summarized in Table 1.

Hedetniemi et al. [11] presented two uniform algorithms (all of the individual processes run the same code) for the dominating set (DS) and the minimal dominating set (MDS) problems. The algorithms work for any connected graph and assume a central daemon (only one process can execute an atomic step at one time). The main idea of the first algorithm is to partition the set of nodes into two disjoint sets, such that each set is dominating. The algorithm for the dominating set (DS) problem stabilizes in linear time ($O(n)$ steps) under a central daemon. The second algorithm calculates a MDS. The main idea of this algorithm is that it allows a node to join the set S, if it has no neighbor in S. On the other hand, a node that is already a member of S, and has a neighbor that is also a member of S, will leave the set if all its neighbors are not pointing to it. Thus, after stabilization the set S will be a MDS. The algorithm for the minimal dominating set (MDS) problem stabilizes in $O(n^2)$ steps under a central daemon.

Goddard et al. [7] gave a self-stabilizing algorithm working on the minimal total dominating set (MTDS) problem. A set is said to be a *total dominating set* if every node is adjacent to a member of it. The authors assume globally unique identifiers for the nodes and a central daemon in [7]. The algorithm uses a mechanism of pointers similar to the one used by the previous algorithm. So, a node i will point to its neighbor having the minimum identifier if i has no neighbor in the set under construction the S. On the other hand, if a node i has more than one neighbor in the set then i will point to null; otherwise i will point to its unique neighbor that is a member of the set S. The algorithm allows a node to join the set S if some neighbor is pointing to it, and to leave the set S otherwise. So after stabilization, the set S will become an MTDS.

Recently, Goddard et al. [8] proposed another uniform self-stabilizing algorithm for finding a minimal dominating set (MDS) in an arbitrary graph under a distributed daemon (a distributed daemon selects a subset of the system processes to execute an atomic step at the same time). The main idea of their

algorithm is that it uses a Boolean variable to determine whether a node is a member of the MDS or not, and an integer to count a node's neighbors that are members of the MDS. The algorithm allows an undominated node that has smaller identifier than any undominated neighbor to join the set under construction. On the other hand, a node leaves this latter set if it is not the unique dominator of itself nor any of its neighbors. The algorithm stabilizes in $O(n)$ steps.

On the other hand, some self-stabilizing algorithms have been proposed in the k-domination case. Kamei and Kakugawa [16] presented two uniform algorithms for the minimal k-dominating set (MKDS) problem in a tree. The first algorithm allows a node to join the set under construction S if it has fewer than k neighbors in S, and to leave the set S if it has more than k neighbors in S. The first algorithm works for a central daemon. Based on this idea, in the second algorithm, a node having more than k neighbors in the set under construction S will first make a request to leave S, and then leaves the set S only if its identifier is the smallest among all the neighbors requesting to leave S. So, after stabilization the set S will become a minimal k-dominating set (MKDS). The second algorithm works under a distribute daemon. The time complexity of the two algorithms are both $O(n^2)$ steps.

Huang et al. [13] presented a self-stabilizing algorithm to find a minimal 2-dominating set (M2DS) in an arbitrary graph. The algorithm allows a node to join the set under construction S if it has fewer than 2 neighbors in S, and to leave the set S if it has more than 2 neighbors in S. The algorithm works under a central daemon, with liner time complexity. Huang et al. also [14] presented another self-stabilizing algorithm to find a minimal 2-dominating set (M2DS) in an arbitrary graph. The algorithm assumes globally unique identifiers for the nodes and works under a distributed daemon. The algorithm allows a node to join the set under construction if it is dominated by fewer than two nodes and none of its neighbors having smaller identifier is in the same situation. Also, a node may leave the set under construction if it is dominated by more than two nodes, and all of its neighbors are either in the set under construction or dominated by more than two nodes.

In 2012, we presented a uniform self-stabilizing algorithm for finding a minimal KDS (MKDS) that works in general graphs under a central daemon [23]. We use a Boolean flag x indicating whether the node is in the constructed set D or not and an integer variable $X(i)$ for counting $i's$ neighbors in D. The algorithm allows a node i to join the set D (the value $x(i) = 1$) under construction if it is dominated by fewer than k nodes in D (R1). Also, a node i may leave the set under construction the set D if it is dominated by more than k nodes (R2). The time complexity of our algorithm in general graphs is $O(n^2)$ steps.

In this paper we design a self- stabilizing algorithm for MKDS (called Algorithm MKDS) under a distributed daemon. The algorithm allows a node i to join the set D (the value $d_i = 1$) under construction if its pointer points to itself and its neighbors which have IDs smaller than its are all dominated at least k times ($R1$). Also, a node i may leave the set under construction the set D if as

far as it can tell all of its neighbors not in D are dominated at least k times by $D - \{i\}$. The time complexity of our algorithm in an arbitrary system graph is $O(n^2)$ steps.

The algorithms we compared in this section are summarized in Table 1. As we can see, the first four self-stabilizing algorithms are for single domination (1-dominating set or total dominating set); and the algorithms for k-dominating set by Kamei et al [16] are just considering in a tree graph; the algorithms by Huang et al. [13,14] are for 2-dominating set. And the self-stabilizing algorithm for MKDS in [23] works under a central daemon. Our Algorithm MKDS is the first work using a distributed daemon approach to discuss the MKDS problem in general networks.

Table 1. Algorithms for dominating set

Reference	Output	Required topology	Self-stabilizing	Daemon	Complexity
Hedetniemi et al. [11]-1	DS	Arbitrary	Yes	Central	$O(n)$ steps
Hedetniemi et al. [11]-2	MDS	Arbitrary	Yes	Central	$O(n^2)$ steps
Goddard et al. [7]	MTDS	Arbitrary	Yes	Central	
Goddard et al. [8]	MDS	Arbitrary	Yes	Distributed	$O(n)$ steps
Kamei et al. [16]-1	MKDS	Tree	Yes	Central	$O(n^2)$ steps
Kamei et al. [16]-2	MKDS	Tree	Yes	Distributed	$O(n^2)$ steps
Huang et al. [13]	M2DS	Arbitrary	Yes	Central	$O(n)$ steps
Huang et al. [14]	M2DS	Arbitrary	Yes	Distributed	
Wang et al. [23]	MKDS	Arbitrary	Yes	Central	$O(n^2)$ steps
Algorithm MKDS	MKDS	Arbitrary	Yes	Distributed	$O(n^2)$ steps

6 Conclusions and Future Work

In the above, we have successfully solved the extension problem, "To find a self-stabilizing algorithm for the minimal k-dominating set problem in general networks under the distributed daemon model". We also verify the correctness of the proposed algorithm and give a proof that the worst case convergence time of the algorithm from any arbitrary initial state is $O(n^2)$ steps where n is the number of nodes in the network.

An immediate extension of this work is to find if it is possible to enhance the stabilization time to $O(nlgn)$ steps. Another future research topic is to attempt to find a proper upper bound size of a k-dominating set in an arbitrary network.

References

1. Datta, A.K., Larmore, L.L., Devismes, S., Heurtefeux, K., Rivierre, Y.: Self-stabilizing small k-dominating sets. International Journal of Networking and Computing 3(1), 116–136 (2013)
2. Dijkstra, E.W.: Self-stabilizing systems in spite of distributed control. ACM. Commun. 17(11), 643–644 (1974)

3. Dijkstra, E.W.: A simple fixpoint argument without the restriction to continuity. Acta. Inf. 23(1), 1–7 (1986)
4. Dolev, S.: Self-Stabilization. MIT Press, Cambridge (2000)
5. Gairing, M., Hedetniemi, S.T., Kristiansen, P., McRae, A.A.: Self-stabilizing algorithms for $\{k\}$-domination. In: Huang, S.-T., Herman, T. (eds.) SSS 2003. LNCS, vol. 2704, pp. 49–60. Springer, Heidelberg (2003)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
7. Goddard, W., Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K.: A self-stabilizing distributed algorithm for minimal total domination in an arbitrary system graph. In: 17th International Symposium on Parallel and Distributed Processing, pp. 485–488. IEEE Press, Nice (2003)
8. Goddard, W., Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K., Xu, Z.: Self-stabilizing graph protocols. Parallel Processing Letters 18(1), 189–199 (2008)
9. Guellati, N., Kheddouci, H.: A survey on self-stabilizing algorithms for independence, domination, coloring, and matching in graphs. Journal of Parallel and Distributed Computing 70(4), 406–415 (2010)
10. Hedetniemi, S.M., Hedetniemi, S.T., Kennedy, K.E., McRae, A.A.: Self-stabilizing algorithms for unfriendly partitions into two disjoint dominating sets. Parallel Processing Letters 23(1) (2013)
11. Hedetniemi, S.M., Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K.: Self-stabilizing algorithms for minimal dominating sets and maximal independent sets. Computer Mathematics and Applications 46(5-6), 805–811 (2003)
12. Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K.: Linear time self-stabilizing colorings. Information Processing Letters 87(5), 251–255 (2003)
13. Huang, T.C., Chen, C.Y., Wang, C.P.: A linear-time self-stabilizing algorithm for the minimal 2-dominating set problem in general networks. Inf. Sci. Eng. 24(1), 175–187 (2008)
14. Huang, T.C., Lin, J.C., Chen, C.Y., Wang, C.P.: A self-stabilizing algorithm for finding a minimal 2-dominating set assuming the distributed daemon model. Computers and Mathematics with Applications 54(3), 350–356 (2007)
15. Li, M., Sun, X., Wang, H., Zhang, Y., Zhang, J.: Privacy-aware access control with trust management in web service. World Wide Web 14(4), 407–430 (2011)
16. Kamei, S., Kakugawa, H.: A self-stabilizing algorithm for the distributed minimal k-redundant dominating set problem in tree network. In: 4th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 720–724. IEEE Press, Chengdu (2003)
17. Kim, D., Zhang, Z., Li, X., Wang, W., Wu, W., Du, D.: A Better Approximation Algorithm for Computing Connected Dominating Sets in Unit Ball Graphs. IEEE Trans. Mob. Comput. 9(8), 1108–1118 (2010)
18. Lana, J.K., Changa, G.J.: Algorithmic aspects of the k-domination problem in graphs. Discrete Applied Mathematics 161(10-11), 1513–1520 (2013)
19. Manne, F., Mjelde, M., Pilard, L., Tixeuil, S.: A new self-stabilizing maximal matching algorithm. Theoretical Computer Science 410(14), 1336–1345 (2008)
20. Sun, X., Wang, H., Li, J., Zhang, Y.: Satisfying Privacy Requirements Before Data Anonymization. Comput. 55(4), 422–437 (2012)
21. Thai, M.T., Wang, F., Liu, D., Zhu, S., Du, D.: Connected dominating sets in wireless networks with different transmission ranges. IEEE Trans. Mob. Comput. 6(7), 721–730 (2007)

22. Wang, H., Zhang, Y., Cao, J.: Effective collaboration with information sharing in virtual universities. IEEE Transactions on Knowledge and Data Engineering 21(6), 840–853 (2009)
23. Wang, G., Wang, H., Tao, X., Zhang, J.: A self-stabilizing algorithm for finding a minimal K-dominating set in general networks. In: Xiang, Y., Pathan, M., Tao, X., Wang, H. (eds.) ICDKE 2012. LNCS, vol. 7696, pp. 74–85. Springer, Heidelberg (2012)
24. Yong, J., Bertion, E.: Replacing Lost or Stolen E-Passports. IEEE Computer 40(10), 89–91 (2007)
25. Yong, J., Shen, W., Yang, Y.: Special Issue on Computer-Supported Cooperative Work: Techniques and applications. Inf. Sci. 179(15), 2513–2514 (2009)
26. Yong, J., Yan, J., Huang, X.: WFMS-based Data Integration for e-Learning. In: 10th International Conference on CSCW in Design, pp. 1361–1366. IEEE, Nanjing (2006)

Logistic Regression Bias Correction for Large Scale Data with Rare Events

Zhen Qiu[1], Hongyan Li[1], Hanchen Su[1], Gaoyan Ou[2], and Tengjiao Wang[2]

[1] State Key Laboratory on Machine Perception Department of Intelligent Science School of EECS, Peking University, Beijing 100871, China
{qiuzhen,lihy,suhanchen}@cis.pku.edu.cn
[2] Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing 100871, China
ougaoyan@126.com, tjwang@pku.edu.cn

Abstract. Logistic regression is a classical classification method, it has been used widely in many applications which have binary dependent variable. However, when the data sets are imbalanced, the probability of rare event is underestimated in the use of traditional logistic regression. With data explosion in recent years, some researchers propose large scale logistic regression which still fails to consider the rare event, therefore, there exists bias when applying their models for large scale data sets with rare events. To address the problems, this paper proposes LRBC method to correct bias of logistic regression for large scale data sets with rare events. Empirical studies compare LRBC with several state-of-the-art algorithms on an actual ad clicking data set. It demonstrates that LRBC method is able to exhibit much better classification performance, and the distributed process for bias correction also scales well.

Keywords: logistic regression, large scale, rare event, bias correction.

1 Introduction

Rare events are often of great interest and significance, their frequency commonly ranges from 0.1% to less than 10%. However, when they do occur, it represents some aspects of reality. Most of the significant events in several areas are rare events, examples abound in fraudulent credit card transactions [1], telecommunication equipment failures [2], oil spills [3], international conflicts [4], state failures [5], train derailments [6], rare events in a series of queues [7] and other rare events. It is important to study rare events in the context of data mining as they can make a lot of sense when correctly classified.

Logistic regression is a classical classification method that has been widely used in many applications. Due to the data explosion in recent years, some researchers have proposed large scale logistic regression models, but they fail to consider a rare event as the dependent variable. Their models become ineffective when directly applied to large scale data sets with rare events. Firstly, they underestimate the probability of rare events for they tend to be bias towards the

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 133–144, 2013.

less important majority class. Secondly, existing data collection strategies may significantly increase the data collection cost and they are not very helpful to detect the rare events. Thirdly, they fail to consider the bias estimate specific to rare events so that they cannot receive an accurate estimate and classified results.

This paper proposes LRBC(*Logistic Regression Bias Correction*) for improving the classification performance with large scale data sets which contain rare events. The method corrects bias after estimating parameters of logistic regression.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 proposes the novel method LRBC for classifying the large scale data sets with rare events. Section 4 describes the data sets and experiment settings. Section 5 shows our experiment results. Section 6 concludes the research and gives directions for future studies.

2 Related Work

Logistic regression [8] [9] is a classification method which has been widely used in many applications. However, when the data sets are imbalanced, it tends to be biased towards the majority class, the probability of rare event is underestimated[10].

To deal with the problem that rare event is underestimated, some researchers propose a few methods. [11] mentions how to correct the bias of logistic regression, but it does not consider the collinearity among the features. [12] proposes a robust weighted kernel logistic regression. However, their methods are designed for small data sets. Thus it may fail to handle large scale data sets.

Due to the rapid increase in the amount of information, there are some existing researches on large scale logistic regression, such as [13] [14] [15] [16]. They proposed some algorithms, but all of them did not consider rare events, their random strategies for collecting examples are also inefficient for data sets with rare events.

There are hardly any researches which consider both rare events and large scale data sets when they apply logistic regression. However, LRBC aims to correct the bias of logistic regression when faced with large scale data sets with rare events.

3 The Proposed Method

This section proposes LRBC method, the flow diagram is shown in Fig. 1. Firstly, we preprocess training samples. Secondly, we estimate the parameters of logistic regression with the large scale preprocessed samples. Thirdly, we correct the bias through distributed process in MapReduce framework.

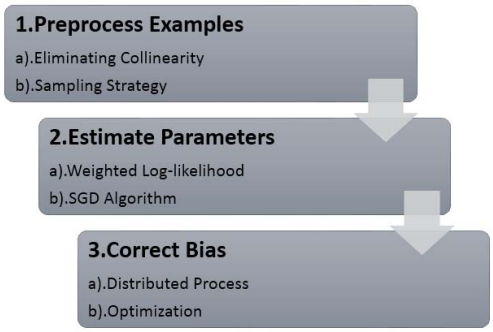


Fig. 1. The flow diagram of LRBC

3.1 Problem Definition

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a data matrix where m is the number of examples and n is the number of features, and y be a binary outcomes vector. For every example $\mathbf{x}_i \in \mathbb{R}^n$ (a row vector in $\mathbf{X}$), where $i = 1, \ldots, n$, the outcome is either $y_i = 1$ or $y_i = 0$. Let the examples with outcomes of $y_i = 1$ belong to the rare events, and the examples with outcomes $y_i = 0$ belong to the common events. The goal is to classify the example $\mathbf{x}_i$ as rare or common.

As mentioned, logistic regression is a widely used binary classification method, the probability π_i when $y_i = 1$ is computed as the function:

$$\pi_i = \frac{1}{1 + \exp\left(-\beta \cdot \mathbf{x}_i\right)} \tag{1}$$

with

$$\beta = [\beta_0, \beta_1, \ldots, \beta_n] \qquad \mathbf{x}_i = [1, x_1, \ldots, x_n]$$

β is estimated by means of MLE(*maximum likelihood estimation*). The goal of LRBC is to correct the bias of β after estimating parameters for large scale data sets with rare events.

3.2 Preprocessing Examples

Eliminate Collinearity. When there is collinearity among the features in $\mathbf{X}$, it will lead the failure of bias correction mentioned in section 3.4.

PCA(*Principal Component Analysis*) method can be applied to eliminate collinearity of the matrix $\mathbf{X}$, it retains the important information through translating original features into the principal component(new features).

In addition, finding the relationship between parent class and child class of features is also useful. For example, one feature x_i represents that the example belongs to *Beijing*, and another feature x_j represents that the example belongs to *China*, because *Beijing* is the capital of *China*. If one example has x_i, it contains x_j undoubtedly. Ranking all the child class by importance, then eliminate the child class which rates poorly until there is no collinearity.

Sampling Strategy. Since existing data collection strategies of logistic regression are inefficient for rare event data, this paper propose the following sampling strategy: Selecting on y by collecting all examples for which $y_i = 1$ and a random selection with a ratio $\eta \in (0, 1)$ for $y_i = 0$. It drastically changes the optimal trade-off between more examples and better variables. Indeed, Many examples which belong to $y_i = 0$ in the data set contain little information. Then recording two fractions which are defined:

$$\overline{\omega}_0 = \frac{1-\tau}{1-\sigma} \quad \overline{\omega}_1 = \frac{\tau}{\sigma} \tag{2}$$

where τ represents the fraction of rare events examples in the original data sets, and the observed fraction of rare events in the samples is expressed as σ. $\overline{\omega}_0$ and $\overline{\omega}_1$ will be used in section (3.3) and (3.4).

3.3 Estimate Parameters

There exist differences in the sample(σ) and original(τ) fractions of ones, LRBC proposes to maximize the weighted log-likelihood instead of the traditional log-likelihood:

$$\ln L(\beta|y) = \overline{\omega}_1 \sum_{y_i=1} \ln \pi_i + \overline{\omega}_0 \sum_{y_i=0} \ln(1-\pi_i) \tag{3}$$

where π_i is defined in Eq.(1), $\overline{\omega}_0$ and $\overline{\omega}_1$ are defined in Eq.(2). The weighted log-likelihood represents that keeping all the rare events and collecting some common events by a ratio, then restore the ratio in the equation. It is shown in is shown in Fig.(2).

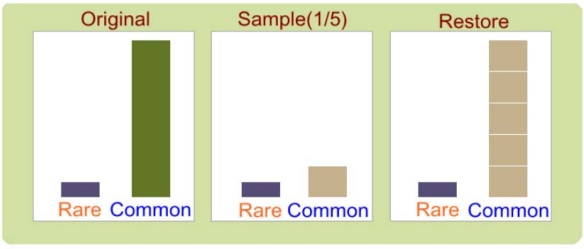


Fig. 2. Weighted Log-likelihood

The negative weighted log-likelihood is as follow:

$$J(\beta) = -\frac{\ln L(\beta|y)}{\overline{\omega}_1} = -(\sum_{y_i=1} \ln \pi_i + \frac{\overline{\omega}_0}{\overline{\omega}_1} \sum_{y_i=0} \ln(1-\pi_i)) \tag{4}$$

In order to estimate the parameter β, LRBC use the stochastic gradient descent algorithm to minimize $J(\beta)$. In each iteration, the stochastic gradient

descent algorithm chooses one training example x_i to update β_{i+1} according to the following formula:

$$\beta_{i+1} = \beta_i - \alpha \frac{\partial}{\partial \beta_i} J(\beta) = \beta_i - \alpha[(1 + \frac{\overline{\omega}_0}{\overline{\omega}_1}) y_j \pi_i - (y_j - \frac{\overline{\omega}_0}{\overline{\omega}_1} \pi_i)] x_j^i \tag{5}$$

where α is the step size, $\frac{\partial}{\partial \beta_i} J(\beta)$ denotes the gradient with respect to β_i.

Algorithm 1 shows the detailed procedure.

Algorithm 1. SGD algorithm for large scale data

1: Input:the step size $\alpha > 0, \{x_1, \ldots, x_m\}, T, \beta_0 = 1$
2: **for** i = 1 to T **do**
3: Draw $j \in 1 \ldots m$ uniformly at random
4: $\beta_{i+1} = \beta_i - \alpha[(1 + \frac{\overline{\omega}_0}{\overline{\omega}_1}) y_j \pi_i - (y_j - \frac{\overline{\omega}_0}{\overline{\omega}_1} \pi_i)] x_j^i$
5: **end for**
6: **return** β

3.4 Correct Bias

LRBC follows the methodology proposed by [11] to correct the bias:

$$bias(\widehat{\beta}) = (X'WX)^{-1} X'W\xi \tag{6}$$

where $\widehat{\beta}$ comes out from Section(3.3) and the bias vector is $bias(\widehat{\beta})$, and $W = diag\widehat{\pi}_i(1 - \widehat{\pi}_i)\omega_i$, $\omega_i = \overline{\omega}_1 Y_i + \overline{\omega}_0(1 - Y_i)$

Theorem 1. *X is the matrix which has no collinearity through Section(3.2), X' is the transposed matrix of X, $X'WX$ becomes a positive definite matrix.*

Proof. *Since $(X'WX)' = X'(X'W)' = X'WX$, $X'WX$ is a symmetry matrix. X has no collinearity among features, $rank(X) = n$, when having arbitrary n-dimensional column vector $a \neq 0$, $Xa \neq 0$, since W is a positive definite matrix, $(Xa)'W(Xa) > 0$, therefore, for arbitrary $a \neq 0$:*

$$a'(X'WX)a = (a'X')W(Xa) = (Xa)'W(Xa) > 0 \tag{7}$$

$X'WX$ is a positive definite matrix which is invertibility.

let Q_{ii} be the ith diagonal element of matrix Q:

$$Q = X(X'WX)^{-1}X' \tag{8}$$

$$\xi_i = 0.5 Q_{ii}[(1 + \omega_i)\pi_i - \omega_i] \tag{9}$$

the bias-corrected estimator is:

$$\widetilde{\beta} = \widehat{\beta} - bias(\widehat{\beta}) \tag{10}$$

Distributed Process. There are many matrix multiplications in the process of bias correction, when faced a large scale data set which produces large matrixes, single node becomes less practical within limited computing and storage resources. We apply distributed process by the framework of MapReduce, which is a software framework introduced by Google to support distributed computing on large scale data sets on clusters of computers, to deal with large matrix multiplications.

The process of calculating Eq.(6) in parallel is decomposed into six steps, as shown in Fig. 3.

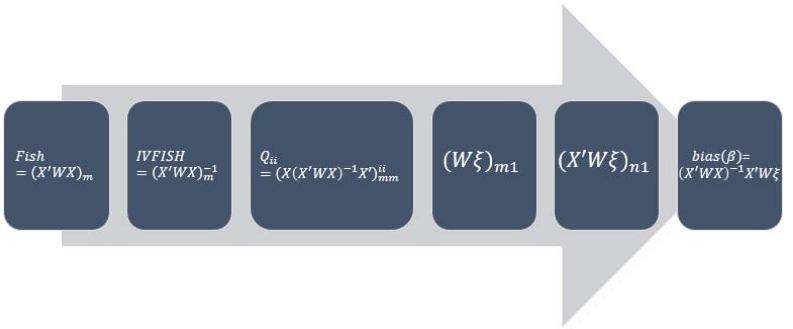


Fig. 3. Calculation Process

In the first step, W is a diagonal matrix, we simplify the matrix operation:

$$Fish_{ij} = (X'WX)_{ij} = \sum_{k=1}^{m} x_{ki}x_{ki}w_k \tag{11}$$

The MapReduece process is shown in in Fig. 4(a).

In the second step, due to the number of features is relatively little, n is small in general, therefore there is no need to use Mapreduce framework to obtain the inverse matrix, we just use common method to determine $(X'WX)^{-1}$.

In the third step, we simplify the matrix operation:

$$Q_{ii} = \sum_{j=1}^{n}\sum_{k=1}^{n} x_{ij}x_{ik}ivfish_{jk} \tag{12}$$

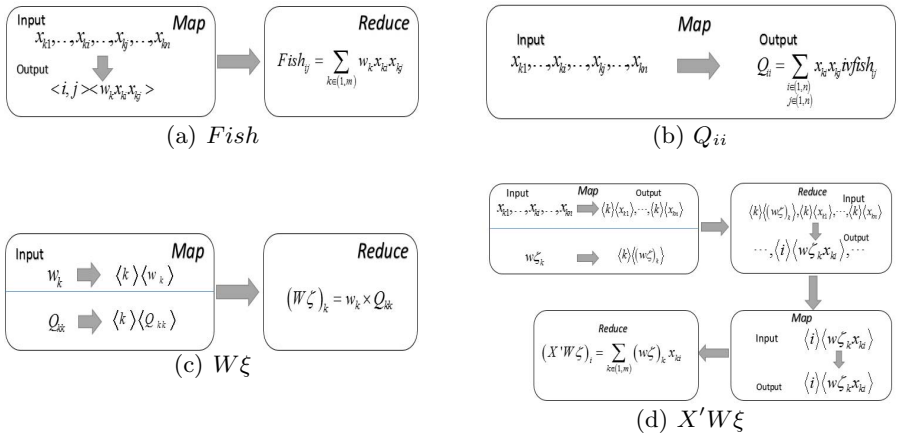
just one map job is enough, which is shown in Fig. 4(b).

In the fourth step, W is a diagonal matrix, and ξ is a matrix which has only one column, the MapReduce process is shown in Fig. 4(c).

In the fifth step, we simplify the matrix operation:

$$(X'W\xi)_i = \sum_{k=1}^{m} x_{ki} \times w\xi_k \tag{13}$$

and use two map jobs and two reduce jobs, which are shown in Fig. 4(d).



(a) $Fish$ (b) Q_{ii} (c) $W\xi$ (d) $X'W\xi$

Fig. 4. MapReduce Process

In the last step, the characters of input matrixes are similar to ones in the fifth step, we apply the same distributed process which belongs to the fifth step.

Optimization. In order to accelerate the running time, we optimize the distributed process by applying a performance model [17] with a task pipeline, it tunes the number of mappers and reducers. The model concludes under the situation that the system capacity is M mappers and R reducers; each mapper has a constant overhead C_1; each reducer has a constant overhead C_2; network transfer rate is V_n; the size of intermediate data is S', the best scheduling plan is to set X mappers and Y reducers where $X = M\sqrt{\frac{C_2}{C_1} + \frac{S'}{C_1 R V_n}}$ and $Y = R$. In addition to this, making the most of CPU capacity in clusters through paralleling the program in map jobs and reduce jobs. According to above work, it can further accelerates the overall running time of distributed process.

4 Experimental Setup

4.1 Data Set

The data set used in this study is derived from an ad clicking data set of one influential Internet company. The probability of an ad gets clicked is minimal, it may be one in ten thousand, the event that an certain ad gets clicked is a rare event undoubtedly. This data set collects 8,834,751 ad clicking examples which contain 412 features, and each example has a binary variable. Of these examples, 24,234 examples belong to the minority class, it represents that the ad gets clicked, all of their variables are 1, while the rest examples belong to the majority class, all of their variables are 0. We collect 90% of the minority class

and the majority class respectively to be training samples, and the rest are used to be test samples.

4.2 Data Preprocessing

At first, eliminating collinearity among 412 features by finding the relationship between the parent class and child class which is mentioned in Section(3.2), then remain all the rare event examples of the train samples, and collect the common event examples by different ratios which is varied in the range of {1e-2,2e-2,...,1e-1}.

4.3 Experiment Environment

The experiments are conducted on a single node(CPU Inter(R)Core(TM)i3-2120M@3.30GHz, Memory 8GB, Hard Disk 1T) with Operation System of WIN7, and a cluster of 27 nodes (Master node&Slave node:CPU AMD Opteron Process @2.6GHz, Memory 48GB, Hard Disk 8T, Network Interface Gigabit Ethernet), with Operation System of SUSE Linux Enterprise Server 11 and Hadoop 0.20.3 as MapReduce implementation.

5 Experiment Results

This section evaluates the performance of LRBC method with two experiments. In the first experiment, we show the classification results of our method and two state-of-the-art methods, In the second experiment, we examine the scalability of the distributed process for bias correction mentioned in Section(3.4).

5.1 Performance Result

In the first experiment, we compare three methods, all of them deal with logistic regression with large scale data sets. LRBC: the method discussed in Section(3), COL: conservative online learning method[13], TRON: the trust region Newton method[16]. However, COL and TRON don't consider rare events.

For LRBC, we empirically set η in the range of {1e-2,2e-2,...,1e-1}. For COL, we choose Auxiliary method, and set $h(z) = ln(\gamma + e^{-z})$ with parameter γ in the range of {1,2,...,10}. For TRON, the regularization parameter C is varied in the range of {1,2,...,16}.

It is important to check the prediction ability, Table 1 shows the accuracy of the classifier learned from the training examples, as well as the training time. For brevity, we only show the partial results around the best parameters for each method. As can be seen, with suitable parameters, the accuracy of LRBC is far higher than COL and TRON, and the training time of LRBC is longer. However, relative to the much higher accuracy, the time cost is in acceptable range.

In addition, we draw ROC diagrams with the above classified results of each method, which is shown in Fig.(5). We also calculate AUC of each method in

Table 1. Accuracy and Training Time of three methods

	LRBC			**COL**		**TRON**	
	η=**0.01**	η=**0.02**	η=**0.1**	$\gamma = 2$	$\gamma = 7$	$C = 2$	$C = 16$
Accuracy	86.72%	87.61%	88.02%	46.34%	41.48%	32.12%	38.34%
Training Time(s)	20,103	20,828	21,023	18,034	18,523	17,349	18,124

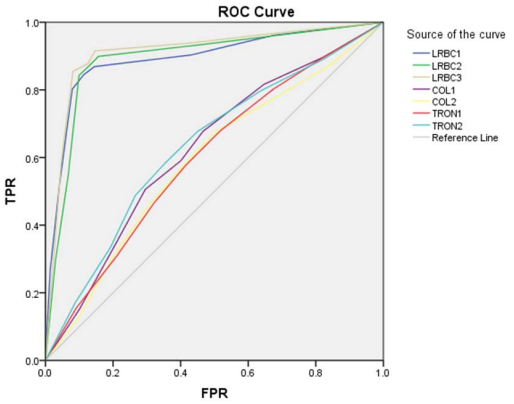


Fig. 5. Comparison of ROC curves

Table 2. AUC of each method

	LRBC			**COL**		**TRON**	
	η=**0.01**	η=**0.02**	η=**0.1**	$\gamma = 2$	$\gamma = 7$	$C = 2$	$C = 16$
AUC	0.893	0.892	0.912	0.624	0.591	0.601	0.634

the ROC curve, the result is shown in Table 2. As can be seen, AUC of LRBC is larger than COL and TRON, and all the AUC of LRBC in this experiment are 0.9 or so, it means that LRBC has a very good performance. We can also find that when η=0.01 or η=0.02 or η=0.1 of LRBC, AUC is almost at the same, it reveals that more examples which belong to common data sets contain little information.

5.2 Scalability Result

This paper propose distributed process to correct bias, this experiment is conducted to examine the scalability of distributed method. When the date is large scale, there exist many large matrixes, single node becomes less practical or cant conduct large matrix operations due to limited resources. For this reason, distributed process offers a solution to handle large matrix operations.

We measure and compare the speedup of distributed process with default setup and optimized setup to verify the scalability using the training data set with $\eta = 0.1$ which has the best performance in the first experiment. LRBC on single node fails with out of memory error, therefore we used 3 nodes as the baseline to measure the speedup of 3/9/15/21/27 nodes in this experiment.

In our cluster environment, parameters are listed as follows: M=27, R=27 (each slave node with 1 mapper and 1 reducer), Vn=10.8M/s, C1=3.3s, C2=3.2s and S'6600M. Under these parameters configuration, we can calculate the optimized setup for X mappers and Y reducers, and we parallel the program in map jobs and reduce jobs, compared with default setup(M mappers and one reducer). The speedup of using different number of nodes is showed in Fig.(6). It shows that distributed method can achieve increasing speedup when more

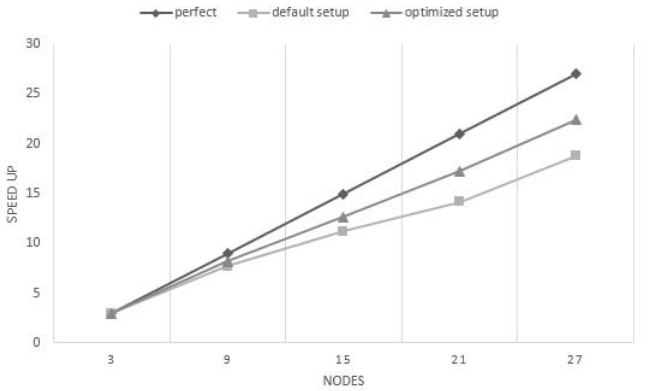


Fig. 6. Speedup of distributed process

Table 3. Execution time for bias correct

Nodes	Default setup	Optimized setup	Time saving
3	12060s	10800s	10.4%
9	8340s	6780s	18.7%
15	6060s	4920s	18.8%
21	4740s	3840s	19.0%
27	4080s	3300s	19.2%

nodes are added into the system, although there lies a gap between perfect linear speedup and distributed method speedup, which is expected due to the increase time spending in network communication over the cluster. Besides, Fig.(6) shows that distributed method with optimized setup achieves higher speedup than that with default setup, and confirms that the optimization can improve the overall performance. From the view of time in Table 3, the optimization can reduce the execution time by about 10% to 20%.

6 Conclusion

In this paper, we emphasize the importance of rare events, and propose LRBC method to correct the bias of logistic regression with large scale data sets containing rare events. Experimental results show that LRBC exhibits better performance when compared with two state-of-the-art methods, and distributed process for bias correction scales well on actual large scale data sets.

For LRBC, it needs more work to reduce training time. In the future, we also plan to apply our method to more areas which have large scale data sets containing rare events.

Acknowledgments. This work was supported by Natural Science Foundation of China (No.60973002 and No.61170003), the National High Technology Research and Development Program of China (Grant No. 2012AA011002), National Science and Technology Major Program (Grant No. 2010ZX01042-002-002-02, 2010ZX01042-001-003-05).

References

1. Chan, P.K., Stolfo, S.J.: Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: KDD, vol. 1998, pp. 164–168 (1998)
2. Weiss, G.M., Hirsh, H.: Learning to predict extremely rare events. In: AAAI Workshop on Learning from Imbalanced Data Sets, pp. 64–68 (2000)
3. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30(2-3), 195–215 (1998)
4. King, G., Zeng, L.: Explaining rare events in international relations. International Organization 55(3), 693–715 (2001)
5. King, G., Zeng, L.: Improving forecasts of state failure. World Politics 53(4), 623–658 (2001)
6. Quigley, J., Bedford, T., Walls, L.: Estimating rate of occurrence of rare events with empirical bayes: A railway application. Reliability Engineering & System Safety 92(5), 619–627 (2007)
7. Tsoucas, P.: Rare events in series of queues. Journal of Applied Probability, 168–175 (1992)
8. Lee, S.I., Lee, H., Abbeel, P., Ng, A.Y.: Efficient l1 regularized logistic regression. In: Proceedings of the National Conference on Artificial Intelligence, vol. 21, p. 401. AAAI Press, MIT Press, Menlo Park, Cambridge (2006)
9. Minka, T.P.: A comparison of numerical optimizers for logistic regression. Unpublished draft (2003)
10. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1), 7–19 (2004)
11. McCullagh, P., Nelder, J.A.: Generalized linear models. Monographs on statistics and applied probability, vol. 37. Chapman Hall, London (1989)
12. Maalouf, M., Trafalis, T.B.: Robust weighted kernel logistic regression in imbalanced and rare events data. Computational Statistics & Data Analysis 55(1), 168–183 (2011)

13. Zhang, L., Jin, R., Chen, C., Bu, J., He, X.: Efficient online learning for large-scale sparse kernel logistic regression. In: AAAI (2012)
14. Liu, J., Chen, J., Ye, J.: Large-scale sparse logistic regression. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 547–556. ACM (2009)
15. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale bayesian logistic regression for text categorization. Technometrics 49(3), 291–304 (2007)
16. Lin, C.J., Weng, R.C., Keerthi, S.S.: Trust region newton method for large-scale logistic regression. The Journal of Machine Learning Research 9, 627–650 (2008)
17. Yang, X., Sun, J.: An analytical performance model of mapreduce. In: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 306–310. IEEE (2011)

An Automatical Moderating System for FML Using Hashing Regression

Peichao Zhang and Minyi Guo

Shanghai Key Laboratory of Scalable Computing and Systems,
Shanghai Jiao Tong University, Shanghai, China
starforever00@gmail.com, guo-my@cs.sjtu.edu.cn

Abstract. In this paper we propose a novel machine learning application on a funny story sharing website for automatical moderation of newly submitted posts based on their content and metadata. This is a challenging task due to the limitation of a machine to understand a joke and the fact that the content of each post is quite short. We collect all the posts of the website using a web crawler, and then extract the features of the posts with the help of some natural language processing (NLP) tools. Finally we utilize a regression model based on approximate nearest neighbor (ANN) search to predict the number of votes for a given post to achieve the goal of determining its quality. Hashing techniques are used to address the curse of dimensionality issue and also for its fast query speed and low storage cost. The experiment shows that our system can achieve a satisfactory performance using various hashing methods.

Keywords: hashing, regression, NLP.

1 Introduction

1.1 Machine Learning and Hashing

The last decade has witnessed the rapid growth and spread of machine learning techniques. It is thought as a big step towards enabling a machine to behave like a human with intelligence. By applying various machine learning algorithms, many processes can be done automatically with few or no human interventions, which is often difficult to achieve using traditional deterministic algorithms. A majority of machine learning algorithms rely on some training data. The key idea is to learn the rule from these training data to optimize some objective function defined by context. With the help of the training data and the properly defined objective function, the algorithms are able to figure out the best rule to use automatically. This is one major aspect in which the machine learning algorithms differ from the traditional deterministic algorithms. An inventor of a machine learning algorithm is not required to know the exact logics within the algorithm, leading to a sense of intelligence. [1]

Nearest neighbor search is one of the fundamental problems in machine learning field. It exists broadly in pattern recognition, information retrieval, data

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 145–156, 2013.

mining and computer vision. In an nearest neighbor search problem, every item is represented as a feature vector. For a given query item, the goal of the problem is to find the item closest to the query, as measured by some distance functions such as Euclidean distance, Mahalanobis distance, and Hamming distance. In practice, it is not necessary to find the exact nearest neighbor for every possible query. This introduces the approximate nearest neighbor (ANN) search problem, which generally results in improved speed and memory saving. [2, 3]

Despite the improvement of ANN search, however, it is still time-consuming to perform an exhaustive search for each new query. The storage cost will also be high for feature vectors of very high dimensions. Besides that, when the dimensions grows large, the distances between the items become indistinguishable, making it difficult for most of the ANN based methods to find the expected items. This is known as the curse of dimensionality. Hashing technology is proposed with the aim to address the above issues [4]. The basic idea of hashing is to map high dimensional data points to low dimensional binary codes with the pair-wise distance similarities preserved. By reducing to feature space of low dimensions, the curse of dimensionality issue can be effectively avoided, and the storage cost gets decreased dramatically as well. Additionally, searching for the items within a fixed Hamming distance to a given query can be done in constant or sub-linear time [5], which makes the ANN search efficient for large dataset.

1.2 About FML

FML[1] is a web collection of funny stories. It is a place for people to publish their own awkward experiences and share their feelings with others. With an intention for people to feel better about their lives through realizing the fact that everybody has its bad day, the website has drawn much attention and is becoming popular.

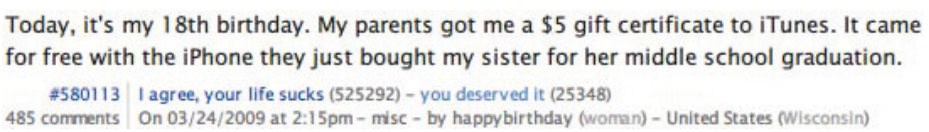


Fig. 1. An example FML post

Figure 1 gives an example FML post. The content of the post is limited to be quite short, like many microblog websites such as Twitter. All the posts are written in English. As depicted in the second line, the users can comment on each post and vote for its goodness or badness. There are also various associated attributes shown in the third line, including time stamp, category, author name, and location.

[1] `http://www.fmylife.com/`

To publish a post, a user needs firstly submits the post to the website. The post is then presented to some other users who volunteer to check the quality of the post and make decision for its acceptance. Such process is called moderation. Only when the post gets accepted will it be shown on the website and available for all other people to view. From the IDs of the published posts we can infer that the total number of submitted posts is about 20 million. Performing moderation on so many posts is a huge amount of work. Thus, we want to automate this process with the help of machine learning techniques. Particularly, an ANN based regression model is used to predict the numbers of votes for the posts based on their features. The predicted numbers of votes measure the quality of the posts and help to decide whether they should be published.

1.3 Contribution

The contributions of this paper are given as follows:

- A dataset from FML is collected with all the published posts.
- With the help of some NLP tools, the feature vectors of the posts are properly extracted from the dataset.
- An automatical moderating system is built using ANN based regression and hashing technology.
- The experiment shows that the performance of the proposed system is satisfactory.

2 The Automatical Moderating System for FML

2.1 Overview

Figure 2 gives an overview of the automatical moderating system for FML. Firstly, the web pages in the website are collected by the web crawler, which recognizes all the posts in the HTML documents and converts them to post objects suitable for later manipulation. Secondly, the post objects are passed to the dict generator, through which a word dictionary is built for feature construction. Thirdly, based on the word dictionary, the feature extractor constructs the feature vectors for the posts. Fourthly, the obtained feature vectors are reduced to low dimensional binary codes by the hash coder with preserved pair-wise similarities. Finally, given a query post in its binary codes form, the predictor gives the predicted votes for the query using the observed votes of the posts in the training data. The details of each component are revealed in the following subsections.

2.2 Web Crawler

Similar to a microblog website, all the posts in the FML website are organized into separate pages with several posts on each page. There are totally 1889 pages on the website, with about 13 posts per page. We send an HTTP GET

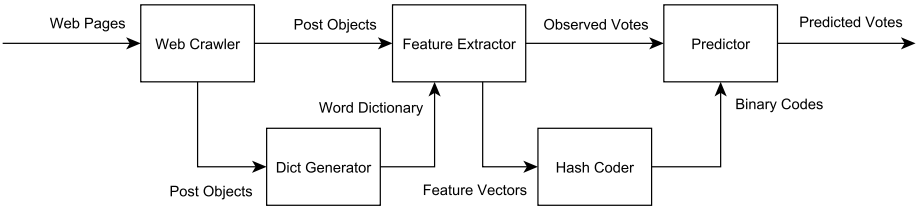


Fig. 2. The overview of the system

request to each page's URL to download all the pages from the website. To avoid being blocked by the server for sending too much requests during an interval, the web crawler pauses for 3 seconds between every requests. The response from the server for each URL is an HTML document. We use lxml[2] to traverse the DOM of the HTML document and extract all the information of the posts with the help of some regular expression matchers. The extracted post information are stored in a local SQLite[3] database as post objects for further manipulation. Table 1 shows all the extracted properties associated with a post object. Some properties are optional and are not available in every post object. At the end, we collected all the published posts from the website with 24,332 post objects in total.

Table 1. Post object properties

Property	Type	Description
content	Unicode string	The content of the post.
num_agree	Integer	The number of good votes on the post.
num_deserve	Integer	The number of bad votes on the post.
num_comment	Integer	The number of comments on the post.
datetime	Datetime	The submission time stamp of the post.
category	String	The category of the post (8 predefined value).
author	Unicode string	(Optional) The user name of the author.
gender	String	(Optional) The gender of the author.
country	String	(Optional) The country the author comes from.
region	String	(Optional) The region in the country.

2.3 Dict Generator

The dict generator is responsible for the generation of the word dictionary, which is a collection of meaningful English words needed by the feature extractor to generate content feature from the post object. We use nltk[4], a leading NLP

[2] `http://lxml.de/`
[3] `http://www.sqlite.org/`
[4] `http://nltk.org/`

toolkit, to process the content text. The word dictionary is built using the following tools contained in the toolkit.

Tokenizer. A tokenizer divides a given text into individual words and punctuations, called tokens. The tokens act as the basic units for all further NLP tasks. For English text, using whitespace to separate words works most of the time. Though some special rules needs to be applied to split standard contractions, like "don't" to "do n't". This can be properly handled by Penn Treebank tokenizer [6].

Stemmer. A English word may have different forms depending on how it's used in a sentence. For example, the word "get" can also be "got", "gotten", and "getting". The purpose of a stemmer is to remove the morphological affixes from a word and turn it to its base form. We use the snowball stemmer [7], a sequel to the famous Porter Stemmer [8], to perform the stemming task. It can correctly transform the regular variants of a word to its base form. For words with irregular variants, it is hard to stem them using a rule-based stemmer. However, the number of such words is quite limited and they are often meaningless to be safely omitted.

Table 2. Selected POS tags

Tag	Description	Tag	Description
JJ	Adjective	RBR	Adverb, comparative
JJR	Adjective, comparative	RBS	Adverb, superlative
JJS	Adjective, superlative	VB	Verb, base form
NN	Noun, singular or mass	VBD	Verb, past tense
NNS	Noun, plural	VBG	Verb, gerund or present participle
NNP	Proper noun, singular	VBN	Verb, past participle
NNPS	Proper noun, plural	VBP	Verb, non-3rd person singular present
RB	Adverb	VBZ	Verb, 3rd person singular present

POS Tagger. With all the tools described above, we construct the word dictionary using the selected tokens in its base form. A part-of-speech (POS) tagger identifies the role of a token in its belonging sentence, such as noun, verb, adjective, etc. We use the POS tagger to filter out those meaningless tokens such as "we", "in", "the", and the punctuations. In this way only the tokens that can better describe the semantic information of a post are included. This also has the side effect of removing most of the stop words since they seldom take a meaningful part in a sentence. The Stanford POS tagger [9] is used to perform this task. Table 2 shows the list of selected POS tags to construct the word dictionary.

We count the number of occurrences of each token in the dataset. Only the tokens that appear at least 5 times are included. As a result, there are 5,156 tokens in the constructed word dictionary.

2.4 Feature Extractor

The feature extractor is used to generate feature vectors for the post objects. The feature vectors act as the descriptors of the post objects in the following regression step. We extract 4 different kinds of features from the properties of a post object to capture its characteristics.

Content Feature. The content feature grasps the semantic information of the posts. It is extracted in our system in a relatively simple way using the word dictionary constructed in Section 2.3. The basic idea is that different kinds of posts may have different usage of words. Particularly, each token in the word dictionary contributes to one dimension of the feature vectors by counting the number of its occurrences in the content of every post. Besides that, another dimension is used to record the total number of tokens for each post. Consequently, we have 5,157 dimensions for the content feature.

Temporal Feature. The time of submission of a post may have effects on its number of votes. This is because the number of online users varies during different time interval of the day. For example, the number of online users at day would be much larger than that at night when most of the people sleep. Therefore a post submitted at day may have more chance to be viewed and voted by users. To model this variance, we use 24 indicator variables to indicate at which time interval of the day a post is submitted. The example post shown in Figure 1 with 2:15pm submission time will have a 1 in its 15th indicator variable and 0s in all others. The similar variances exist between weekday and weekend, and holiday and workday. We use 7 indicators for day of the week, 12 indicators for month of the year, and 366 indicators for day of the year. This results as 409 dimensions in total for the temporal feature.

Location Feature. Some posts in the collected dataset have optional country/region properties associated. Inspired by the observation of the temporal feature, the posts come from the same country or region may share more interests than other posts. To model this, we use indicator variables to indicate at which latitude/longitude interval a post comes from. There are 270 indicators in total, with 90 and 180 for latitude and longitude respectively. We use geopy[5] to obtain the latitude/longitude pair from the country/region text through geocoding. The geocoder uses Google Maps V3 engine as a backend to fulfill the request. We cache the results locally for query speedup and also to avoid exceeding the daily usage limit of the engine. For posts without location properties, or with

[5] `https://code.google.com/p/geopy/`

locations which cannot be resolved by the geocoder, their indicator variables are set to be all 0s.

Miscellaneous Feature. Besides the features mentioned above, there are other properties of the post objects that can be utilized to generate feature vectors. This includes the gender of the author, and the category of the post. We use 2 and 8 indicator variables to represent these two properties respectively in a similar way as the temporal and location features.

To sum up, the total number of dimensions of the feature vectors is 6,140. These feature vectors will be used to predict the number of votes in the following process.

2.5 Hash Coder and Predictor

To predict the number of votes for a given query post, we need some amount of posts with known number of votes as the training dataset. Suppose the number of posts in the training dataset is N, the feature vectors of these posts can be represented by a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where $D = 6,140$ is the total number of dimensions of the feature. We also have a vector $\mathbf{v} \in \mathbb{R}^N$ denoting the corresponding numbers of votes for the posts in the training dataset. For a given query q with its feature vector denoted as $\mathbf{x}_q \in \mathbb{R}^D$, the basic idea of regression is to predict its number of votes v_q using the information of the posts in the training dataset which are close to q. The closeness of two posts are often defined between their feature vectors using Euclidean distance, or some other distance functions like Mahalanobis distance.

However, there are some issues in the above approach.

1. The dimensions of the feature are rather high with $D = 6,140$. In such situation, the closeness between the posts becomes indistinguishable no matter which distance function is used. Due to this fact, the numbers of votes cannot be predicted accurately using regression. This problem is known as the curse of dimensionality.
2. To find the closest posts to a query as measured by Euclidean distance, we need a linear scan of all the posts in the training dataset. For dataset with large N, the time cost for regression would be too high to accept.
3. The cost to store the posts in the training dataset would also be high due to the large values of N and D.

With these concerns, we count on hashing technology to reduce high dimensional feature vectors $\mathbf{X}$ to low dimensional binary codes $\mathbf{B} \in \mathbb{R}^{N \times Q}$, where Q is the dimension of the binary codes. We use Hamming distance to measure the closeness between two binacy codes, which is the number of positions at which they differs. The original closeness measured by Euclidean distance between feature vectors should be preserved in the generated binary codes. Specifically, when two posts have a low Euclidean distance between their feature vectors, the Hamming distance between their binary codes should also be low. With this

property hold, we can predict v_q for a given query q in the same way as the original regression except in the binary code space.

By using hashing, the curse of dimensionality can be effectively avoided thanks to the fact that Q is usually much smaller than D. The storage cost also gets lowered dramatically since storing a binary bit is 64 times cheaper than storing a double precision floating number. Using Hamming distance, for a given query in the binary codes form, its closest posts can be retrieved in constant or sub-linear time. This makes the regression quite efficient on datasets with large N.

There are many different kinds of hashing methods for the generation of the binary codes. They can be roughly divided into three categories: data independent, unsupervised data dependent, and supervised data dependent [10]. Data independent methods generate the binary codes without using any training data. Unsupervised data dependent methods learn the binary codes from some training data. And supervised data dependent methods learn the binary codes with some additional supervision information.

In this paper we use some classic hashing methods from each of the three categories to learn their differences in various aspects. This includes: locality-sensitive hashing (LSH) [2–4], anchor graph hashing (AGH) [11], iterative quantization (ITQ) [12], sequential projection learning for hashing (SPLH) [13], and kernel-based supervised hashing (KSH) [10].

For supervised data dependent methods such as SPLH and KSH, a similarity matrix $\mathbf{S} \in \{0,1\}^{N\times N}$ is required as the source of the supervision information. With the aim to predict v_q accurately, we define $\mathbf{S}$ as

$$S_{ij} = \begin{cases} 1, & |v_i - v_j| \le T \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where T is a threshold value controlling the sparsity of $\mathbf{S}$.

We use two different settings to predict v_q for a given query q with the binary codes generated by hashing methods.

Radius Bound (RB). In radius bound setting, v_q is computed through averaging over v_i for all items i with their Hamming distances to q no more than a predefined bound B. Concretely, we have

$$v_q = \mathbb{E}_{i:H_q(i)\le B}[v_i], \tag{2}$$

where $H_q(\cdot)$ is the Hamming distance to the query q, and $\mathbb{E}[\cdot]$ is the averaging operator.

K-Nearest Neighbor (KNN). The k-nearest neighbor setting is quite similar to the radius bound setting, except we take the average over the items with the K lowest Hamming distances to the query q. When there are ties, we include all items with the same Hamming distances of the K-th lowest items. In particular,

$$v_q = \mathbb{E}_{i:H_q(i)\le R(q)}[v_i], \tag{3}$$

where $R(q)$ is chosen for each q so that there are at least K items i with $H_q(i) \leq R(q)$.

3 Experiment

We perform the experiment using the dataset consisting of all the posts of the FML website, with 24,332 posts in total. We randomly selected 1000 posts as the validation set, which is used by some hashing methods like KSH to choose their hyper-parameters. We randomly selected another 1000 posts as the query set to evaluate the performance of our system. The rest of the posts are used as the training set for the binary codes generation and the prediction of votes.

The value of T in (1) is set to make every post in the training set to have roughly 200 neighbors in $\mathbf{S}$. For prediction settings, we set $B = 2$ for RB and $K = 5$ for KNN.

The experiment is conducted on a workstation with 24 Intel Xeon CPU cores and 64 GB RAM. All the reported results are averaged over 10 independent runs with different training/validation/query partitions.

3.1 Prediction Accuracy

We use root mean squared error (RMSE) to measure the prediction accuracy performance. For all posts in the query set, suppose the predicted numbers of votes are $\widetilde{\mathbf{v}}$, and the actual numbers of votes are $\mathbf{v}$, RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}(\widetilde{v}_i - v_i)^2}, \tag{4}$$

where N_t is the size of the query set. A lower RMSE means a better prediction accuracy performance.

Figure 3 shows the prediction accuracy of different hashing methods with different prediction settings at different code lengths. For all hashing methods with KNN prediction setting, the changes in the code length have little impact on their prediction accurcy, and the differences in performance between the hashing methods are not significant. On the other hand, the prediction accuracy of some hashing methods (LSH, ITQ, and KSH) with RB prediction setting improves a lot as the code length increases. The overall prediction accuracy performance of all settings is satisfactory given the fact that the number of votes of a post can be as large as 1 million.

3.2 Success Rate

The success rate measures the percentage of successful predictions for quries in the query set. For RB prediction setting, the prediciton would fail for a query if

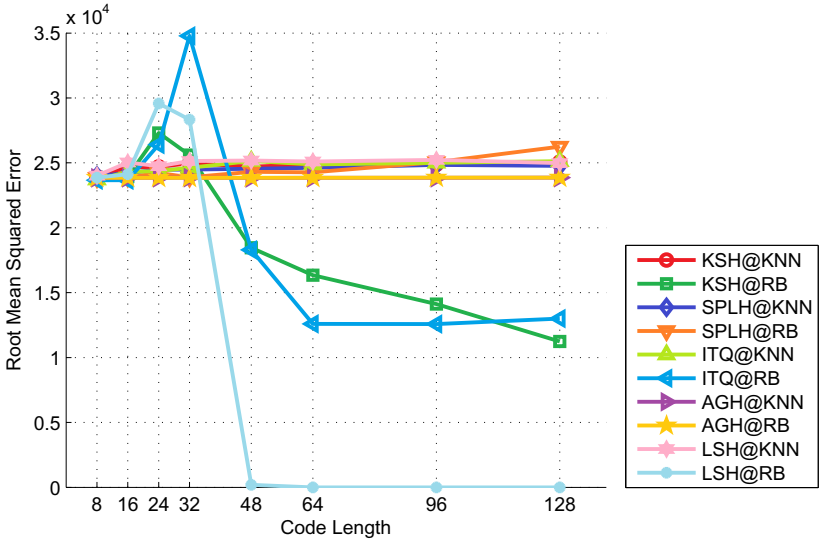


Fig. 3. Prediction accuracy result

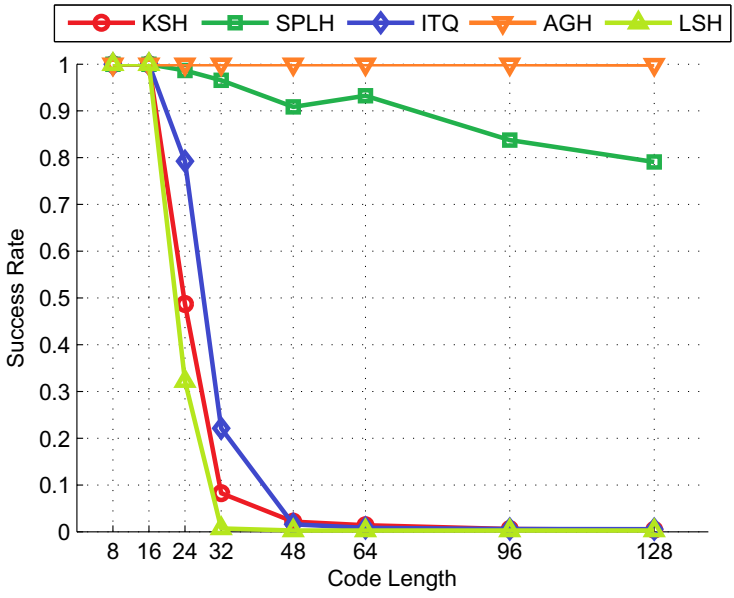


Fig. 4. Success rate result

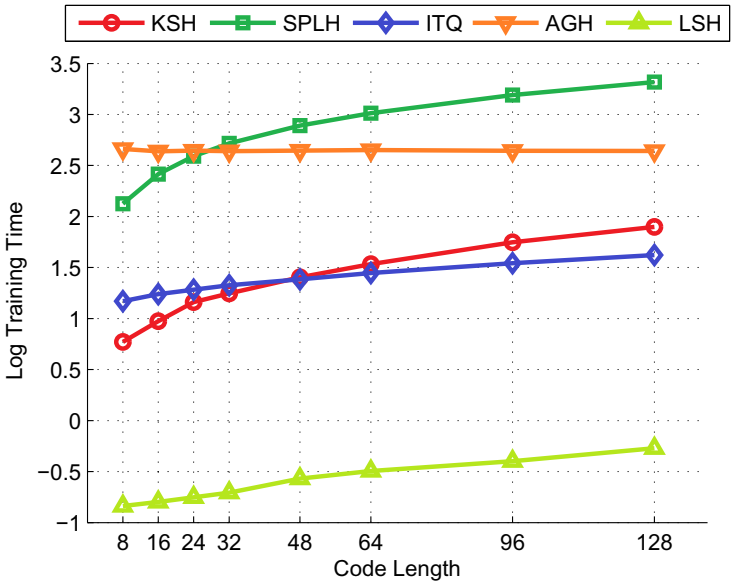


Fig. 5. Computational cost result

there is no item falls within its Hamming distance bound. For KNN prediction setting, the prediction always succeeds.

Figure 4 shows the success rates of different hashing methods with RB prediction setting at different code lengths. For some hashing methods like LSH, ITQ, and KSH, their success rates drops significantly as the code length goes large. The reason is that the overall distances between the items will increase with the code length, thus there will be more chance for a query to have 0 item within its fixed radius bound. Other hashing methods like SPLH and AGH don't suffer much (or any) from the increase of the code length since they have some built-in mechanisms to prevent the above situations.

3.3 Computational Cost

Figure 5 shows the computational costs of binary codes generation for different hashing methods. The reported time costs are in log scale. Quite naturally, the time costs of almost all methods increase with the code length except for AGH due to its independence on the code length. Basically, the data independent hashing methods require the least amount of time for binary codes generation since they do not use any training data. On the contrary, the supervised data dependent hashing methods use both the training data and the supervision information to learn their binary codes, leading to the most amount of computational cost.

4 Conclusion

In this paper we propose an automatical moderating system for FML website using machine learning techniques. We collect the posts of the whole website and build the dataset using different kinds of features extracted by an NLP toolkit. An ANN based regression model with two different settings are applied to predict the numbers of votes for the query posts. We use various hashing methods to address the problems caused by feature vectors of very high dimensions. The experiment results show that the performance of the proposed system is satisfactory. Our system can be easily extended for prediction of other values of interest like the number of bad votes or the number of comments.

Acknowledgement. This work is partially supported by the 863 Program of China (No. 2011AA01A202), Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT) China, Shanghai Excellent Academic Leaders Plan (No. 11XD1402900), and NSFC (Grant No. 60725208, 61003012).

References

1. Bishop, C.M., Nasrabadi, N.M.: Pattern Recognition and Machine Learning. J. Electronic Imaging 16(4), 049901 (2007)
2. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: VLDB, 518–529 (1999)
3. Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: STOC, pp. 604–613 (1998)
4. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: FOCS, pp. 459–468 (2006)
5. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: CVPR (2008)
6. Marcus, M.P., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: Annotating predicate argument structure. In: HLT. Morgan Kaufmann (1994)
7. Porter, M.F.: Snowball: A language for stemming algorithms (2001)
8. Porter, M.F.: An algorithm for suffix stripping. Program: Electronic Library and Information Systems 14(3), 130–137 (1980)
9. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: HLT-NAACL (2003)
10. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: CVPR, pp. 2074–2081 (2012)
11. Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: ICML, pp. 1–8 (2011)
12. Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: CVPR, pp. 817–824. IEEE (2011)
13. Wang, J., Kumar, S., Chang, S.F.: Sequential projection learning for hashing with compact codes. In: Fürnkranz, J., Joachims, T. (eds.) ICML, pp. 1127–1134. Omnipress (2010)

Batch-to-Batch Iterative Learning Control Based on Kernel Independent Component Regression Model

Ganping Li[1], Jun Zhao[1], Fuyang Zhang[1], and Zhizhen Ni[2]

[1] School of Information Engineering, Nanchang University, Nanchang 330031, China
gplee1972@sina.com,
sekel1982@aliyun.com,
zfuyang@126.com
[2] State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China
nzz1987@gmail.com

Abstract. A model-based batch-to-batch iterative learning control (ILC) strategy for batch processes is proposed in this paper. The data-driven model of batch process is developed using kernel independent component regression (KICR) method when the operating data of batch process have a non-Gaussian distribution. The ILC algorithm is derived based on the linearization of the KICR model around the control profile. Applications to a simulated nonlinear batch reactor demonstrate that the proposed ILC strategy can improve process performance from batch to batch when the operating data of batch process follow non-Gaussian distribution. Comparisons between KICR model based and support vector regression (SVR) model based ILC strategies are also made in the simulation. The results show the KICR model based ILC has better performance.

Keywords: Iterative Learning Control, Kernel Independent Component Regression, Batch Process.

1 Introduction

Batch process plays an increasingly important role in industry. Compared with continuous processes, batch processes generally have high nonlinearities and are running in the transient state (there is no steady state in the process). In addition, product quality of batch processes generally cannot be measured online and can only be acquired through laboratory analysis after a process is completed. All these determine the control of batch processes is challengeable.

In the last decade, a method called iterative learning control (ILC) is applied to batch process to obtain high-quality products. ILC is mainly used for the control of repetitive processes. It uses the information of previous run to improve the performance of current run. Since batch processes are of repetitive nature, thus ILC can be applied to batch process to improve the product quality from batch to batch.

By now, many ILC strategies have been proposed for batch process control, such as the ILC combined with model predictive control (MPC) [1-3] and the ILC using

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 157–164, 2013.

artificial neural network (ANN) models [4-6]. Different models are used in those ILC strategies.

In the past decades, data-driven modeling techniques have received a great deal of attentions. These techniques extract the essential information from historical dataset to build process models. However, many of them are linear modeling methods, such as partial least-squares (PLS) and principal components regression (PCR). For a nonlinear process, a linear modeling technique cannot build an accuracy model. In recent years, kernel methods were developed very fast. By mapping the original dataset into a feature space using a kind of kernel functions, nonlinear regression can be conducted from a dataset. Kernel methods provide new strategies for data-based nonlinear modeling. Since batch processes generally are nonlinear processes, some kernel methods such as support vector regression (SVR) and kernel PLS (KPLS) were employed to build the nonlinear models of the batch processes in some ILC strategies[7],[8]. However, for SVR and KPLS, it is assumed that data follow a Gaussian distribution in feature space. When data do not meet the condition, such modeling methods are not so effective.

In the paper, an ILC strategy based on kernel independent component regression (KICR) model is proposed for batch process control. KICR is derived from kernel independent component analysis (KICA) [9], an effective kernel method that has capacity to handle non-Gaussian distributed data. Using the model, the proposed ILC strategy can deal with the control problem when the data of batch process are non-Gaussian distributed. The simulation results show the advantages of the ILC strategy.

2 KICR

The KICA approach developed is essentially KPCA plus ICA. First, training data are whitened and mapped into a feature space as linearly separable as possible. Assuming that the input data matrix $\boldsymbol{X}(m\times n)$ has been normalized, m is the number of variable and n is the number of sample. The nonlinear mapping that maps the original data onto the linear high-dimensional feature space F is defined byΦ , which is $\Phi: R^m \to F, \quad x \mapsto \overline{X}$. Then the covariance matrix of the mapped data can be expressed as:

$$\mathrm{cov}(\Phi(\boldsymbol{X})) = \frac{1}{n}\Phi(\boldsymbol{X})\Phi(\boldsymbol{X})^T \tag{1}$$

Because $\Phi(\cdot)$ is difficult to acquire, “kernel tricks” can be used to evaluate an inner product in the feature space by defining the Gram kernel matrix as follows:

$$k_{ij} = \Phi(\boldsymbol{x}_i)^T\Phi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2}$$

$$\boldsymbol{K} = \Phi(\boldsymbol{X})^T\Phi(\boldsymbol{X}) \tag{3}$$

$$\boldsymbol{k}_x = \Phi(\boldsymbol{X})^T\Phi(\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the *i*th and *j*th sample of **X** respectively, and x is a new sample.

The radial basis function (RBF) $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2) / 2\sigma^2$ is selected as kernel function in order to model the nonlinear correlation structure properly. Additionally, the mapped data should be centered in the linear high-dimensional feature space *F*, which was given as

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{I}_n\boldsymbol{K} - \boldsymbol{K}\boldsymbol{I}_n + \boldsymbol{I}_n\boldsymbol{K}\boldsymbol{I}_n \tag{5}$$

$$\tilde{\boldsymbol{k}}_x = \boldsymbol{k}_x - \boldsymbol{I}_n\boldsymbol{k}_x - \boldsymbol{K}\boldsymbol{I}_1 + \boldsymbol{I}_n\boldsymbol{K}\boldsymbol{I}_1 \tag{6}$$

where $\boldsymbol{I}_n = \frac{1}{n}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{n\times n}$, $\boldsymbol{I}_1 = \frac{1}{n}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$; $\tilde{\boldsymbol{K}}, \tilde{\boldsymbol{k}}_x$ are the centered kernel matrix.

According to the formula (1), the whitening transformation can be deduced as follows [10]:

$$\boldsymbol{z} = \sqrt{n}\boldsymbol{D}^{-1}\boldsymbol{E}^T\tilde{\boldsymbol{k}}_x \tag{7}$$

where $\boldsymbol{D}$ is the diagonal matrix of $\tilde{K}$ and $\boldsymbol{E}$ is the corresponding eigenvector of $\boldsymbol{D}$.

To reserve the useful information and eliminate the noises, PCA method was employed to reduce the dimension of centered kernel matrix by exploiting the fact that the eigenvalue contribution is above δ%. Therefore, we get the reduced Dd and the corresponding Ed. Then formula (7) which gets reduced has the form of

$$\boldsymbol{z} = \sqrt{n}\boldsymbol{D}_d^{-1}\boldsymbol{E}_d^T\tilde{\boldsymbol{k}}_x + \boldsymbol{\epsilon} \tag{8}$$

where $\boldsymbol{z}(m\times 1)$ is the whitened vector of the input vector $\boldsymbol{x}(m\times 1)$, and *d* is the number of reserved eigenvalues, $\boldsymbol{\epsilon}$ is the vector of residuals given by the dimension reduction.

After the mapping and whitening, the reduced kernel independent components $\boldsymbol{S} = [\boldsymbol{s}_1\ \boldsymbol{s}_2\ \ldots\ \boldsymbol{s}_d]$ can be expressed as [11]

$$\boldsymbol{S} = \boldsymbol{W} \cdot \boldsymbol{Z} \tag{9}$$

where $\boldsymbol{Z} = [z_1\ z_2\ \ldots\ z_n]$ is the whitened data matrix of $\boldsymbol{X}$. $\boldsymbol{W}$ is the de-mixing matrix which can be obtained using fix-point ICA algorithm [12].

If the output data matrix $\boldsymbol{Y}(l\times n)$ has also been normalized, a relationship between $\boldsymbol{Y}$ and $\boldsymbol{S}$ in the linear high-dimensional feature space $\boldsymbol{F}$ can be built as follows:

$$\boldsymbol{Y} = \boldsymbol{B}^T \cdot \boldsymbol{S} + \boldsymbol{H} \tag{10}$$

where $\boldsymbol{B}$ is the regression coefficient matrix and $\boldsymbol{H}$ is the residual error matrix.

By using least squares regression method, we have

$$\boldsymbol{B} = (\boldsymbol{S} \cdot \boldsymbol{S}^T)^{-1} \boldsymbol{S} \cdot \boldsymbol{Y}^T \tag{11}$$

This finally results into the following KICR model for function estimation:

$$f(x) = \boldsymbol{A} \cdot K_x = \sum_{i=1}^{n} a_i K(x, x_i) + b \tag{12}$$

where $\boldsymbol{A} = \sqrt{n}\boldsymbol{B}^T \boldsymbol{W} \boldsymbol{D}_d^{-1} \boldsymbol{E}_d = [a_1\, a_2 \cdots a_n]$, b is a bias term.

3 ILC Methodology

The final product qualities of a batch process can be expressed as

$$\boldsymbol{y}(t_f) = \boldsymbol{F}(\boldsymbol{X}_0, \boldsymbol{U}) \tag{13}$$

where $\mathbf{y}(t_f) = [y_1(t_f), y_2(t_f) \cdots y_N(t_f)]$ is a vector of final product qualities at batch end time t_f. $\mathbf{X}_0$ is the initiate condition of the batch process and $\mathbf{U} = [u_1, u_2, \cdots, u_L]^T$ is a vector of control inputs by dividing a batch process into L time segments of equal length. The nonlinear function vector $\boldsymbol{F}(\cdot,\cdot)$ is represented by KICR model.

Based on the KICR model formulated by (12), the optimal control policy $\mathbf{U}$ can be obtained by solving the following optimization problem:

$$\min_{U} J[\boldsymbol{y}(t_f)] \tag{14}$$

s.t. product quality and process constraints

The first order Taylor series expansion of (13) around a nominal control profile can be expressed as

$$\hat{\mathbf{y}}(t_f) = \boldsymbol{F}_0 + \frac{\partial \boldsymbol{F}}{\partial u_1}\Delta u_1 + \frac{\partial \boldsymbol{F}}{\partial u_2}\Delta u_2 + \cdots + \frac{\partial \boldsymbol{F}}{\partial u_L}\Delta u_L \tag{15}$$

The actual final product quality for the kth batch can be written as

$$\mathbf{y}_k(t_f) = \hat{\mathbf{y}}_k(t_f) + \boldsymbol{e}_k \tag{16}$$

where $\mathbf{y}_k(t_f)$ are the actual product qualities and $\hat{\mathbf{y}}_k(t_f)$ are the predicted product qualities at the end of the kth batch respectively, and $\boldsymbol{e}_k$ is the model prediction error.

From (15), the prediction for the kth batch can be approximated using the first order Taylor series expansion based on the (k-1)th batch:

$$\begin{aligned}\hat{\mathbf{y}}_k(t_f) &= \hat{\mathbf{y}}_{k-1}(t_f) + \frac{\partial \boldsymbol{F}}{\partial u_1}\Big|_{U_{k-1}}(u_1^k - u_1^{k-1}) + \frac{\partial \boldsymbol{F}}{\partial u_2}\Big|_{U_{k-1}}(u_2^k - u_2^{k-1}) + \cdots + \frac{\partial \boldsymbol{F}}{\partial u_L}\Big|_{U_{k-1}}(u_L^k - u_L^{k-1}) \\ &= \hat{\mathbf{y}}_{k-1}(t_f) + \boldsymbol{G}_k^T \Delta \mathbf{U}^k\end{aligned} \tag{17}$$

where

$$\Delta \mathbf{U}^k = [\Delta u_1^k \quad \Delta u_2^k \quad ... \quad \Delta u_L^k]^T,$$

$$\boldsymbol{G}_k^T = \left[\frac{\partial \boldsymbol{F}}{\partial u_1}\Big|_{U_{k-1}} \quad \frac{\partial \boldsymbol{F}}{\partial u_2}\Big|_{U_{k-1}} \quad \cdots \quad \frac{\partial \boldsymbol{F}}{\partial u_L}\Big|_{U_{k-1}} \right]^T.$$

The control input can be calculated employing the conventional quadratic objective function:

$$\min_{\Delta \mathbf{U}^k} J_k = \left\| \mathbf{y}_d - \mathbf{y}_{k-1}(t_f) - \boldsymbol{G}_k^T \Delta \mathbf{U}^k \right\|_Q^2 + \left\| \Delta \mathbf{U}^k \right\|_R^2 \tag{18}$$

where $\mathbf{y}_d$ is the objective value, $\boldsymbol{Q}$ is a weighting matrix for the end state errors and $\boldsymbol{R}$ is a weighting matrix for the control effort.

For the unconstrained case, set $\partial J / \partial \Delta \mathbf{U}^k = 0$, an analytical solution to the above minimization can be obtained as

$$\Delta \mathbf{U}^k = (\boldsymbol{G}_k \boldsymbol{Q} \boldsymbol{G}_k^T + \boldsymbol{R})^{-1} \boldsymbol{G}_k \boldsymbol{Q} (\mathbf{y}_d - \mathbf{y}_{k-1}(t_f)) \tag{19}$$

$$\mathbf{U}^k = \mathbf{U}^{k-1} + \Delta \mathbf{U}^k \tag{20}$$

The gradient of model output with respect to **U** and $\boldsymbol{G}$, can be calculated as

$$\boldsymbol{G}_k = \frac{\partial \boldsymbol{F}}{\partial \mathbf{U}}\Big|_{\mathbf{U}_{k-1}} = \sum_{i=1}^{n} a_i \frac{\partial \boldsymbol{K}(\mathbf{U}, \mathbf{U}_i)}{\partial \mathbf{U}}\Big|_{\mathbf{U}_{k-1}} = \sum_{i=1}^{n} -a_i \frac{(\mathbf{U}_{k-1} - \mathbf{U}_i)}{\sigma^2} \exp\left(-\frac{\left\| \mathbf{U}_{k-1} - \mathbf{U}_i \right\|_2^2}{2\sigma^2}\right) \tag{21}$$

where $\boldsymbol{G}_k$ is the gain matrix of the ILC.

4 Simulation Example

A nonlinear batch reactor is chosen to illustrate the performance of the ILC strategy. The reaction system is described as $A \xrightarrow{k_1} B \xrightarrow{k_2} C$, where *A* is the raw material, *B* is the product, and *C* is the by-product. The differential equations describing the batch reactor are given as [13]

$$\begin{aligned} \frac{dx_1}{dt} &= -k_1 \exp(-E_1 / uT_{ref}) x_1^2 \\ \frac{dx_2}{dt} &= k_1 \exp(-E_1 / uT_{ref}) x_1^2 - k_2 \exp(-E_2 / uT_{ref}) x_2 \end{aligned} \tag{22}$$

where x_1 and x_2 are the dimensionless concentrations of *A* and *B* respectively, $u = T / T_{ref}$ is the dimensionless temperature of the reactor and T_{ref} is the reference temperature, the reactor temperature *T* is the control variable. The parameter values of

the batch reactor are $k_1 = 4.0\times10^3$, $k_2 = 6.2\times10^6$, $E_1 = 2.5\times10^3 K$, $E_2 = 5.0\times10^3 K$, $T_{ref} = 348K$. The final time of the process is $t_f = 1.0$ h. The initial conditions are $x_1(0) = 1$, $x_2(0) = 0$. The control performance is to reach the objective end-point value $\boldsymbol{y}_d(t_f) = [x_2(t_f)] = [0.6060]$.

The sample time of the batch process is set 3s. In order to generate the training data to build the KICR model, random changes with uniform distribution are added to the initiate trajectory and data of 30 batch runs are generated to develop the model. Based on the KICR model, we apply the proposed ILC scheme to the batch reactor for 300 batches. The kernel parameter of KICR model was selected as $\sigma = 6.7$, the parameters of ILC were chosen as $Q = 1000$, $R = 0.01I$. The results are shown in Fig. 1. It can be seen from Fig. 1 that the control performance is significantly improved from batch to batch and converge to the target. The total simulation time is 9.141s. For comparison, the SVR model based ILC is also applied to the batch process. Least squares SVR with a RBF kernel is adopted for modeling. The results are shown in Fig. 2. It can be seen from Fig. 2 that the ILC does not converge to the desired value. The total simulation time of this method is 11.216s. Thus KICR model based ILC has higher computation efficiency.

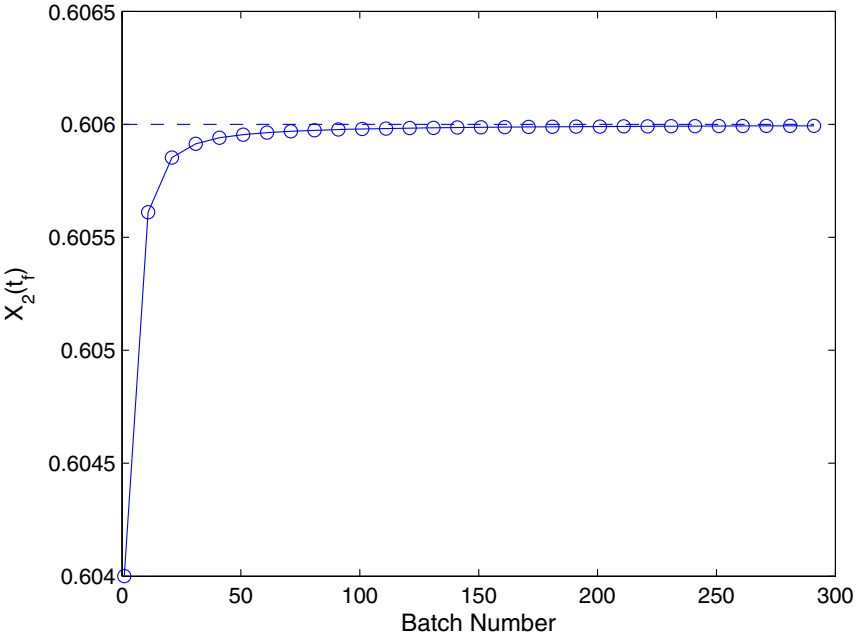


Fig. 1. KICR model based ILC performance

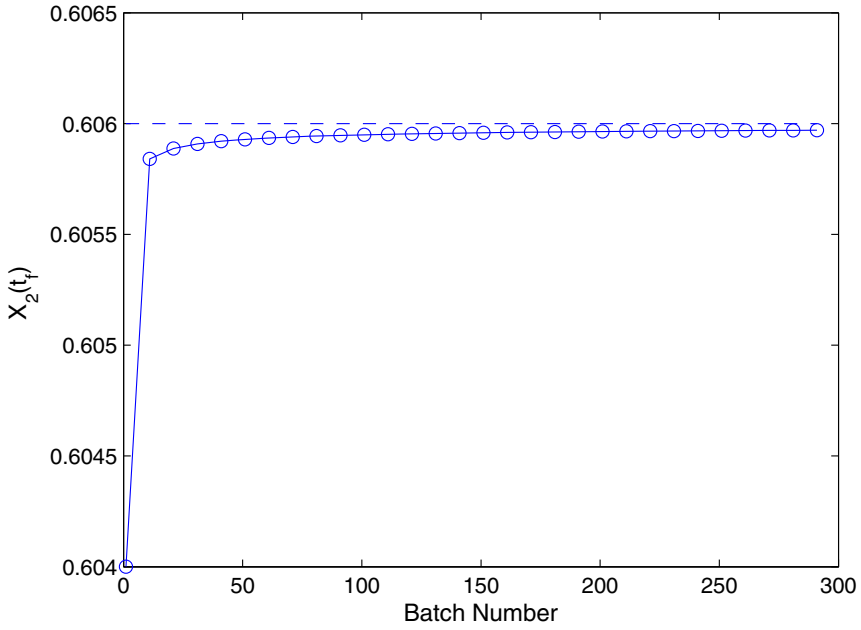


Fig. 2. SVR model based ILC performance

5 Conclusions

A batch-to-batch model-based iterative learning control strategy for the end-point product quality control of batch process is proposed. To address the problem of nonlinearities in batch processes, a nonlinear model for end-point product quality prediction, linearized around the nominal batch trajectories, is identified from process operating data using KICR technique. Based on the linearized KICR model, an ILC law is obtained explicitly by calculating the optimal control profile. Since KICR method is capable of processing non-Gaussian distributed data, thus, when the operating data of batch process are non-Gaussian distributed, the KICR model based ILC strategy is useful. The simulation results demonstrate the effectiveness of the method.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Grant 61064004.

References

1. Lee, K.S., Chin, I.S., Lee, H.J., Lee, J.H.: Model predictive control technique combined with iterative learning for batch processes. AIChE Journal 45, 2175–2187 (1999)
2. Lee, J.H., Lee, K.S., Kim, W.C.: Model-based iterative learning control with a quadratic criterion for time-varying linear systems. Automatica 36, 641–659 (2000)

3. Lee, K.S., Lee, J.H.: Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables. Journal of Process Control 13, 607–621 (2003)
4. Zhang, J.: Neural network model based batch-to-batch optimal control. In: IEEE International Symposium on Intelligent Control, pp. 352–357. IEEE Press, New York (2003)
5. Zhang, J.: Multi-objective optimal control of batch processes using recurrent neuro-fuzzy networks. In: Proceedings of the International Joint Conference on Neural Networks, pp. 304–309. IEEE Press, New York (2003)
6. Xiong, Z., Zhang, J.: A batch-to-batch iterative optimal control strategy based on recurrent neural network models. Journal of Process Control 15, 11–21 (2005)
7. Liu, Y., Yang, X., Xiong, Z., Zhang, J.: Batch-to-Batch Optimal Control Based on Support Vector Regression Model. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3498, pp. 125–130. Springer, Heidelberg (2005)
8. Zhang, Y.W., Fan, Y.P., Zhang, P.C.: Combining kernel partial least-squares modeling and iterative learning control for the batch-to-batch optimization of constrained nonlinear processes. Industrial Engineering and Chemical Research 49, 7470–7477 (2010)
9. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. Journal of Machine Learning Research 3, 1–48 (2003)
10. Tian, X., Zhang, X., Deng, X., Chen, S.: Multiway kernel independent component analysis based on feature samples for batch process monitoring. Neurocomputing 72, 1584–1596 (2009)
11. Antoni, W., Ishak, D.M.: Nonlinear robust regression using kernel principal component analysis and R-estimators. International Journal of Computer Science Issues 8, 75–82 (2011)
12. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 10, 626–634 (1999)
13. Xiong, Z., Zhang, J.: Batch-to-batch optimal control of nonlinear batch processes based on incrementally updated models. IEE Proceedings - Control Theory and Applications 151, 158–165 (2004)

Deep Architecture for Traffic Flow Prediction

Wenhao Huang[1], Haikun Hong[1], Man Li[2], Weisong Hu[2],
Guojie Song[1,⋆], and Kunqing Xie[1]

[1] Key Laboratory of Machine Perception, Ministry of Education,
Peking University, Beijing, 100871, China
[2] NEC Labs, China
rubio8741@gmail.com, gjsong@pku.edu.cn

Abstract. Traffic flow prediction is a fundamental problem in transportation modeling and management. Many existing approaches fail at providing favorable results duo to 1)shallow in architecture;2)hand engineered in features. In this paper, we propose a deep architecture consists of two parts: a Deep Belief Network in the bottom and a regression layer on the top. The Deep Belief Network employed here is for unsupervised feature learning. It could learn effective features for traffic flow prediction in an unsupervised fashion which has been examined effective for many areas such as image and audio classification. To the best of our knowledge, this is the first work of applying deep learning approach to transportation research. Experiments on two types of transportation datasets show good performance of our deep architecture. Abundant experiments show that our approach could achieve results over state-of-the-art with near 3% improvements. Good results demonstrate that deep learning is promising in transportation research.

Keywords: Deep Learning, Deep Belief Nets, Traffic Flow Prediction.

1 Introduction

Traffic flow prediction is an important work in transportation management. Without accurate traffic flow prediction, none of intelligent transportation systems could work well. Many research attentions have been focused on this subject in recent years. Existing traffic flow prediction approaches could be divided into three categories. 1)Time-series approaches[20][12]. These approaches such as ARIMA model [20] focus on finding patterns of temporal variation of traffic flow and use that for prediction. 2)Probabilistic approaches[18][21][16]. Modeling and forecasting of traffic flow is done from probabilistic perspective. 3)Nonparametric approaches[15][1][2][14]. Researchers demonstrated that nonparametric approach generally performs better due to their strong ability to capture the indeterministic and complex nonlinearity of traffic time series. Some representative methods are artificial neural networks (ANN)[15][9], support vector regression (SVR)[1] and local weighted learning (LWL)[14].

⋆ Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 165–176, 2013.

Many systems mainly possess two failings. 1)They are shallow in architectures. For neural network approaches, the architecture is usually designed only consisting of one single hidden layer. For other methods such as time-series approaches, linear architecture is often preferred. 2)Tedious and error-prone hand-engineered features are needed for some approaches. They require prior knowledge on specific domain for feature extraction and selection.

In this paper, we attempt to define a deep architecture for traffic flow prediction that learns features without any prior knowledge. This is achieved by training a Deep Belief Network (DBN) which is based on work of Geoffrey Hinton et.al[4][5]. The key idea is using greedy layer-wise training with stacked Restricted Boltzmann Machines (RBMs) and followed by fine-tuning. The DBN is used for unsupervised feature learning in traffic flow prediction. Upon them, a regression layer could be added for supervised training. Contribution of this paper could be concluded in three aspects.

- To the best of our knowledge, this work is the first attempt to introduce deep learning approaches into transportation research. Characteristic of transportation system such as huge amount in data and high dimensions in features would make deep learning a promising method for transportation research.
- Deep architecture could doing prediction from a network perspective. It could integrate input data from all observation points in several time intervals together.
- Experiments show good performance of deep architecture for traffic flow prediction. It could achieve near 3% improvements comparing with state-of-the-arts.

Rest of the paper is structured as follows. Section 2 presents background knowledge of traffic flow prediction and deep belief network. In Section 3 we introduce our deep architecture for traffic flow prediction. Section 4 gives experimental results of our approach and analysis of these results. Finally, conclusion and future works are described in Section 5.

2 Background

2.1 Traffic Flow Prediction

Traffic flow prediction has long been regarded as a critical problem for intelligent transportation systems. It aims at estimating traffic flow of a road or station in next several time intervals to the future. Time intervals are usually defined as short-term intervals varying from 5 minutes to 30 minutes. For operational analysis, the Highway Capacity Manual (TRB 2000) [10] suggests using a 15 minutes time interval. Two types of data are usually used in traffic flow prediction. One is data collected by sensors on each road such as inductive loops. The task is predicting traffic flow on each road or segment. The other type of data is collected at begin and end of a road. For example, we would get a card from

the toll station entering a highway and have to turn it back when we leave the highway in some countries. This kind of data is referred as entrance-exit station data. Despite predicting traffic flow on each road, another task is forecasting traffic flow in each station especially the exit station. Traffic flow of i^{th} observation point (spatial id, road or station) at t^{th} time interval is denoted as $f_{i,t}$. At time T, task is to predict traffic flow $f_{i,T+1}$ at time $T+1$ based on traffic flow sequence $F = \{f_{i,t} | i \in O, t = 1, 2, \ldots, T\}$ in previous where O is the full set of observation points. Some tasks may focus on predicting traffic flow of next several time intervals from $T+1$ to $T+n$ as well.

Traffic flow prediction consists of two steps: feature learning and predicting model learning. **Feature learning** learns a feature representation model g which extracts and selects most representative features from traffic flow sequence F of all stations in previous. After feature learning, traffic flow sequence is transformed into feature space $g(F) \rightarrow X$. Prediction task $f_{i,T+1}$ could be represented as Y. In some approaches, feature learning is ususally hand-engineered. Some important factors for transportation such as speed, volume of flow and density are calculated from raw data and used as features for prediction. Moreover, in most of approaches, only time-series features are employed for prediction. For example, ARIMA model only selects previous traffic flow of a specific point j:

$$\forall y_j = f_{j,T+1}, x_j = \{f_{j,t} | t = T, T-1 \ldots, T-m+1\} \tag{1}$$

where m is time step in ARIMA model. It do not use any extra data except previous traffic flows for the task j self. **Predicting model learning** is a supervised learning problem. Given feature and task pairs obtained from history traffic flow $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, it learns a predicting model $\widehat{y} = h(x)$ that minimizing loss function

$$L(y, \widehat{y}) = \frac{1}{2}(y - \widehat{y})^2. \tag{2}$$

As introduced, a variety of predicting models are established in previous studies.

2.2 Deep Belief Network

Recent works on deep learning have demonstrated that deep sigmoidal networks could be trained layer-wise to produce good results for many tasks such as image and audio classification[6][4][7]. Idea of deep learning is first using large amount of unlabeled data to learn feature by pre-training a multi-layer neural network in an unsupervised way and then using labeled data for supervised fine-tuning to adjust learned features slightly for better prediction.

Deep Belief Network is the most common and effective approach among all deep learning models. It is a stack of Restricted Boltzmann Machines each having only one hidden layer. The learned units activations of one RBM are used as the "data" for the next RBM in the stack. Hinton et al. proposed a way to perform fast greedy learning of DBN one layer at a time[4].

An RBM is an undirected graphical model in which visible variables ($\boldsymbol{v}$) are connected to stochastic hidden units ($\boldsymbol{h}$) using undirected weighted

connections[19]. They are restricted that there are no connections within hidden variables or visible variables. The model defines a probability distribution over $\boldsymbol{v}, \boldsymbol{h}$ via an energy function. Suppose it is a binary RBM, it could be written as:

$$\begin{aligned} &-\log P(\boldsymbol{v}, \boldsymbol{h}) \propto E(\boldsymbol{v}, \boldsymbol{h}; \theta) \\ &= -\sum_{i=1}^{|V|}\sum_{j=1}^{|H|} w_{ij} v_i h_j - \sum_{i=1}^{|V|} b_i v_i - \sum_{j=1}^{|H|} a_j h_j \end{aligned} \tag{3}$$

where $\theta = (\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{a})$ is parameters, w_{ij} is the symmetric weight between visible unit i and hidden unit j while b_i and a_j are their bias. Number of visible and hidden units is represented as $|V|$ and $|H|$. This configuration makes it easy to compute the conditional probability distributions, when $\boldsymbol{v}$ or $\boldsymbol{h}$ is fixed.

$$\begin{aligned} p(h_j|\boldsymbol{v}; \theta) &= sigm(\sum_{i=1}^{|V|} w_{ij} v_i + a_j) \\ p(v_i|\boldsymbol{h}; \theta) &= sigm(\sum_{j=1}^{|H|} w_{ij} h_j + b_i) \end{aligned} \tag{4}$$

where $sigm(x) = \frac{1}{1+e^{-x}}$ is a sigmoid function. The parameters of the model $\theta = (\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{a})$ could be learned using contrastive divergence [3] effectively.

Then we could stack several RBMs together into a DBN. The key idea behind trianing a DBN by training a series of RBMs is that parameters θ learned by an RBM define both $p(\boldsymbol{v}|\boldsymbol{h}, \theta)$ and prior distribution $p(\boldsymbol{h}|\theta)$[11]. Therefore, probability of generating visible variables could be written as:

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\theta) p(v|h, \theta) \tag{5}$$

After θ is learned from an RBM, $p(\boldsymbol{v}|\boldsymbol{h}, \theta)$ is kept. In addition, $p(\boldsymbol{h}|\theta)$ could be replaced by consecutive RBM which treats hidden layer of previous RBM as visible data. By this way, it could improve a variational lower bound on the probability of the training data as introduced in [4]. DBN could be used as unsupervised feature learning method if no labels are provided.

3 Learning Architecture

Previous approaches of traffic flow prediction are all shallow in architecture. Instead we advocate a deep architecture in this paper. Training deep multilayered neural network is generally hard because the error gradient would explode or vanish when number of layers is increasing. Recent works on deep learning have made training deep neural network more effective and efficient since Hinton's breakthrough in 2006[4]. Here, we employ a DBN for unsupervised feature learning and add a regression layer above the DBN for traffic flow prediction.

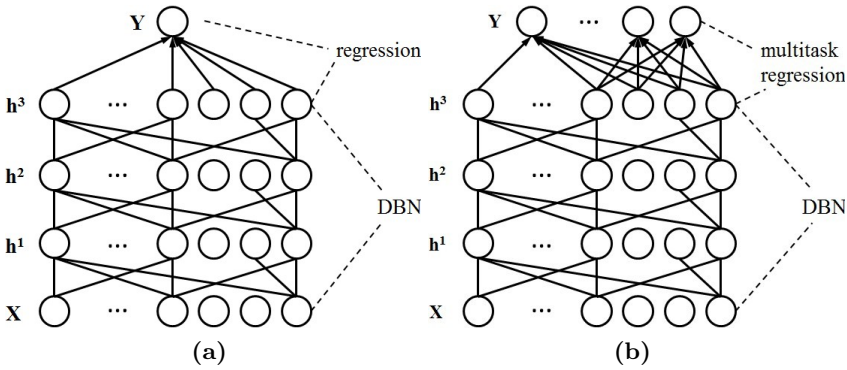


Fig. 1. Deep architecture for traffic flow prediction.(a)A DBN in the bottom for unsupervised feature learning with a sigmoid regression layer at the top for supervised prediction. It is an architecture for traffic flow prediction of a single road.(b)All tasks are trained jointly via multi-task regression.

DBN could learn a more effective feature representation than raw data in an unsupervised way without much prior knowledge. Then we use sigmoid regression at the top layer in our approach so that we could perform supervised fine-tuning on the whole architecture easily. Sigmoid regression layer could also be replaced with other regression models such as support vector regression.

Our deep architecture for traffic flow prediction on a single road or station is summarized in Figure 1(a). Further, we could employ multi-task regression to train all the tasks jointly as shown in Figure 1(b). The input space X is generally the raw data we collected. To do prediction from a network perspective, we let all the observation points be in the input space. Moreover, we take full advantage of traffic flow of previous several time intervals. Therefore, the input space is large ($|O| \times |T|$, where $|O|$ is number of observation points and $|T|$ is number of time intervals). Number of previous time frames k is the only prior knowledge we need to build the predicting model. We tested several values for k and chose the best one through cross validation ($k = 4$ here). We do not apply any artificial feature extraction and selection from raw data except for selection of k. The only pre-prossing work is normalizing traffic flow into [0,1].

Unlike binary RBM as introduced in Section 2.2, we replace it with real-valued units [13] that have Gaussian noise to model traffic flow data. Energy function and conditional probability distributions are given as follows:

$$\begin{aligned} &-\log P(\boldsymbol{v}, \boldsymbol{h}) \propto E(\boldsymbol{v}, \boldsymbol{h}; \theta) \\ &= \sum_{i=1}^{|V|} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j=1}^{|H|} a_j h_j - \sum_{i=1}^{|V|} \sum_{j=1}^{|H|} \frac{v_i}{\sigma_i} h_j w_{ij} \end{aligned} \tag{6}$$

$$p(h_j|\boldsymbol{v};\theta) = sigm(\sum_{i=1}^{|V|} w_{ij}v_i + a_j)$$
$$p(v_i|\boldsymbol{h};\theta) = N(b_i + \sigma_i \sum_{j=1}^{|H|} h_j w_{ij}, \sigma_i^2) \tag{7}$$

where σ is the standard deviation vector of Gaussian visible units and $N(\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ.

Since traffic flows of all the observation points are used as input data, we have to regularize the model for sparsity[8]. We encourage each hidden unit to have a predetermined expected activation by a regularization penalty of the form:

$$\lambda \sum_{j=1}^{|H|} (\rho - \frac{1}{m}(\sum_{k=1}^{m} E[h_j|\boldsymbol{v}^k]))^2 \tag{8}$$

where ρ determines the sparsity and $\boldsymbol{v}^k$ is a sample in training set with total m samples.

4 Experiments and Results

4.1 Experiment Settings

Datasets. Two datasets are used in this study. One benchmark dataset is obtained from the California Freeway Performance Measurement System (**PeMS**)[1]. It is a kind of inductive loop dataset and the task is predicting traffic flow on the road near the loop detector. This system continuously collects loop detector data in real time for more than 8100 freeway locations throughout the state of California. Then the data are aggregated into 5 minutes periods and are accessible on the internet for research. We further aggregate the data into 15 minutes periods as suggested by Highway Capacity Manual. Traffic flow of a road is obtained from averaging all loop detectors in the road. Then we choose roads of top 50 traffic flows for study since roads with large traffic flows cause more attention in transportation research. We average data of loop detectors in the same road to compute traffic flow of a link road and choose roads of top 50 traffic flow for study. Another dataset we employed is from highway system of China (entrance-exit station of highway,**EESH**). In each entrance and exit of highway, there is a station for charging and recording related information. Data are collected in each station and aggregated into 15 minutes periods. Task in EESH is predicting traffic flow in exit stations.

[1] `http://pems.dot.ca.gov/`

Training and Testing Data. Both datasets contain data of totally 12 months while we use data of first 10 months as training set and later 2 months as testing set.

Evaluation Metrics. Mean absolute percent error (MAPE) is used for error measurement. It is computed as:

$$MAPE = \frac{|\widehat{y} - y|}{y} \tag{9}$$

where $\widehat{y}$ is the predicted flow and y is the real value. We could get mean accuracy (MA) where *MA=1-MAPE*. For overall performance evaluation, we use weighted mean accuracy (WMA) which takes traffic flow as weight and it implies the aim of predicting high-flow areas more accurately.

Architectures. There are many parameters we have to define for the deep architecture for traffic flow prediction such as nodes in each layer, layer size, epochs and time intervals (k) of input data as introduced. These parameters are chosen through cross validation only on training set to ensure fairness when comparing with other approaches. In next section, we will further analyze the effect of each parameter to last results.

4.2 Structure of Deep Architecture

Our models contains several parameters to be defined for building the architecture. *Time intervals k* which determines structure of input data ranges from 1 to 16 (15 minutes to 4 hours). We choose *layer size* from 1 layer to 7 layers. For simplicity, *number of nodes* in each layer is set to be the same. It is chosen from $\{16, 32, 64, 128, 256, 512, 1024\}$. *Epochs* of training is also important to learning phase. The model would overfit in training data if number of epochs is too large. We let *epochs* range from 10 to 100 with 10 as a gap. We first choose each parameter randomly from the possible set and then choose the best configuration from 1000 random runs. The best structure we recorded is as follows: *layer size=3, nodes in layers=128, epochs=40* and *time intervals k=4*. Then we test effect of each parameter to our deep architecture while keeping other parameters fixed. In this step, testing set is used to evaluate generalization error. We believe we could find a better parameter configuration using grid search or other heuristic searching methods. However, due to large search spaces, it would be very tedious and computationally unacceptable. Random search in a fixed set is preferred in our experiment. Default task for structure parameter selection is traffic flow prediction on PeMS.

First we examine influences of different network structures. Issue of network size choosing is one of the most typical problem for neural network design. The learning time and the generalization capabilities of the particular neural network model are highly affected by the network size parameter. The result is reported

in Table 1 and Table 2. Table 1 shows the weighted mean accuracy, number of weights and training time with variation of number of layers. In this experiment, number of nodes in each layer is fixed the same (128 here). Performance could be improved with the increase of layers from 1 to 3. More complex structures do not have advantages over a 3-layer structure. Since we only employ 10 months' training data, models with very complex structure would be under fitted. Number of weights and training time demonstrated spatial and temporal complexity of each model. They all increase linearly with the increasing of layers. During the process of computing, we used GPU for acceleration. Training time seems acceptable under setting *number of layers=3*.

Table 1. Effect of number of layers

Layers	WMA	Weights	Time
1	0.846	12800	83s
2	0.875	29184	219s
3	0.897	45568	337s
4	0.889	61952	466s
5	0.881	78336	574s
6	0.863	94720	688s
7	0.837	111104	820s

Table 2. Effect of nodes in a layer

Nodes	WMA	Weights	Time
16	0.812	2112	51s
32	0.843	5248	79s
64	0.881	14592	152s
128	0.897	45568	337s
256	0.891	156672	894s
512	0.886	575488	2544s
1024	0.884	2199552	8549s

Table 2 gives the result of variation number of nodes in each layer. Similarly, 128 nodes in each layer is the best choice. Unlike the result of number of layers, spatial and temporal complexity increases exponentially. More nodes in each layer would cause unnecessary burden for model training and compromise the performance. However, less nodes in each layer may cause the model could not learn representative features. For the consideration of both simplicity and accuracy, the structure of 3 layers with 128 nodes in each layer is used.

Then we investigate effect of *epochs* and *input time intervals* k. Figure 3 shows the curve of accuracy on training set and testing set as a function of number of epochs. With the increase of epochs, error on training set could be improved while the generalization capability does not improve if number of epochs is larger than 40. The model seems overfitted on the data when number of epochs is too large. Apparently, large epoches which would lead large temporal cost is not appropriate in our model though they could improve accuracy on training set in some extent.

For *time intervals* k, a large k would increase size of the first layer which could be seen from number of weights. It is almost a linear increasing trend. But it fails at improving performance after $k = 4$. Average travel time of cars on the road is about 1 hour (4 time intervals). Therefore, traffic flow of each road in 4 time intervals is most related to each other. If k is over 4, more irrelevant inputs would make the complex architecture difficult to learn a good representation. In fact, *time intervals* k for each task (road or station) should be different. For some roads with more long distance cars, a big k may be better. We should choose an appropriate k for each task separately instead of same k for each task for best

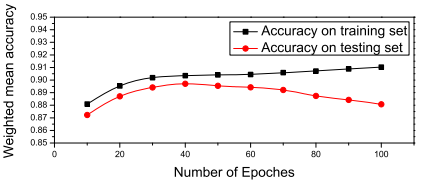


Fig. 2. Effect of epoch times

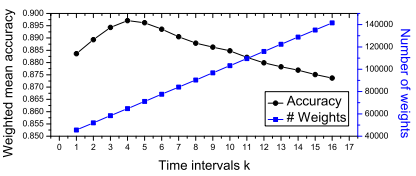
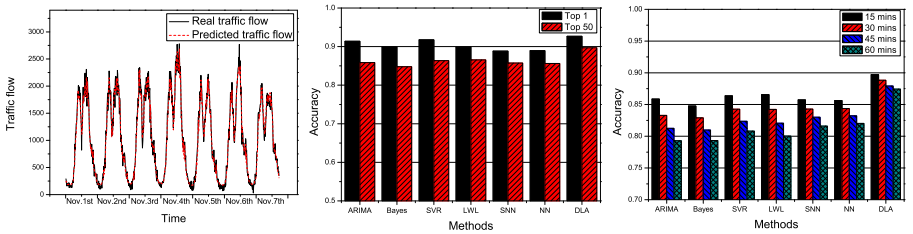


Fig. 3. Effect of input time intervals

overall performance. However, for simplicity of computation, we use a fixed k for each task in this paper.

4.3 Results of Deep Learning Architecture

Here we investigate learning and generalization capabilities of our deep architecture, and compare it with other existing approaches. Several wide-spread methods are employed as comparing approaches in this study. They are ARIMA model[20], Bayesian model[18], SVR model[1], LWL model[14], simple neural network model (SNN) and neural network model (NN). SNN is a one layer neural network. NN is a neural network with the same architecture as our approach (DLA) while it uses backpropagation without pre-training. These models are trained and tested using the same training and testing set as used for deep architectures while input data may be a little bit different.



(a) **(b)** **(c)**

Fig. 4. Performance of our deep architecture for traffic flow prediction and comparison with existing approaches.(a)Predicting results of road with largest flow for a week.(b)Comparison on prediction accuracy on top 1 road and top 50 roads.(c)Comparison of predicting multi-time intervals.

Three tasks are used here to evaluate each method:1)predicting the road with largest traffic flow, 2) predicting roads with traffic flow in top 50, 3)predicting several time intervals in advance. As shown in Figure 4(a), performance of our approach works quite well. Predicted traffic flow and real flow could perfectly match especially in peak time. From Figure 4(b), our deep neural network model without hand-engineered features could outperform all existing

approaches. For the road with largest flow, all the approaches work well in fact. Accuracy is over 90% as reported in many existing researches. DLA could improve the accuracy slightly (about 0.8%). The advantage of DLA is more obvious when we take top 50 roads into account. Existing approaches are not very effective to roads with middle level traffic flow. Due to the ability of non-linear structure and unsupervised feature learning, DLA could provide favorable results in almost all the roads with an improvement on weighted overall accuracy by over 3%. Actually, if we take average mean accuracy into account without weight, improvements are more obvious (over 5%). This is because improvements are bigger when examining only middle flow or small flow roads. Figure 4(c) demonstrated another advantage of our deep architecture. Existing approaches demand that input features should be strongly related with output. Thus they are usually limited in short-term traffic flow prediction. Accuracy would derease a lot when predicting traffic flow of several time intervals in advance. However, DLA could still be robust. Improvements of prediction accuracy would increase from 3% to near 7% when prediction time interval is increased from 15 minutes to 60 minutes. We also examined the cases of 12 hours later. Accuracy of DLA is still over 75% while it of ARIMA is only near 60%. From the results, we could also see that pre-training in deep learning is useful when comparing with random weight initialization as NN model adopted.

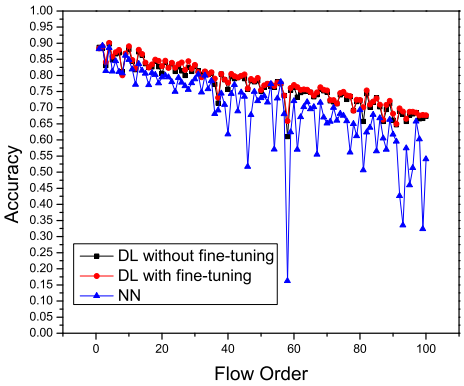


Fig. 5. Detail comparison of deep learning architecture and neural network approach

In EESH, we could obtain similar results. WMA could be improved from 75% to near 80%. Since deep learning is highly related with neural network, we give a detail comparison of deep learning and neural network in Figure 5. In the figure, DL without fine-tuning approach only uses unsupervised pre-training before regression while DL with fine-tuning approach also employs supervised fine-tuning. NN is in the same deep architecture with DL but uses backpropagation for network training. It is clear in the figure that pre-training is useful in model training. Nearly accuracy for all the stations could be improved.

Thus it is kind of universal advantage but not specific one. Fine-tuning could only achieve slight advantages which also indicates that unsupervised pre-training is very effective for weight initialization.

In conclusion, deep learning method is effective in traffic flow prediction. It could imply complex relationship of transportation system. The advantage of unsupervised feature learning would make this approach easier for application. It is promising to apply deep learning into transportation research. Many related transportation problems such as transportation induction and transportation management could employ deep learning method for better result.

5 Conclusions

In this study, we have presented a deep machine learning architecture for traffic flow prediction, implemented as a stack of RBMs in the bottom with a regression layer on the top. The stack architecture in the bottom is a Deep Belief Network and it is effective for unsupervised feature learning. This is the first work of employing deep learning in transportation area. Without hand-engineered feature extraction and selection, our architecture could learn a good representation of features. The top regression layer is used for supervised training. From experiments on two real traffic flow datasets, we demonstrated that our deep architecture could improve accuracy of traffic flow prediction. With limited prior knowledge, it could learn effective feature representations. Result of our approach could outperform state-of-the-art approach with near 3% improvements. It is a promising start of applying deep learning method to transportation research.

There are still many potential works to do of deep learning in transportation research. One is using temporal deep neural networks instead of static networks. Deep learning is traditionally used for static tasks such as image classification. The problem of how to use temporal information in traffic flow prediction would be interesting and valuable to explore. Another possible direction is building a robust prediction system based on deep architecture. In real application, many problems such as data missing and data noise would make the theoretically sound approach not practical[17]. Deep architecture is more robust than other methods due to its complex structure. It is important to use this advantage for building a practical prediction system.

References

1. Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., Han, L.D.: Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications 36(3), 6164–6173 (2009)
2. Clark, S.: Traffic prediction using multivariate nonparametric regression. Journal of Transportation Engineering 129(2), 161–168 (2003)
3. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14(8), 1771–1800 (2002)
4. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)

5. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
6. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1106–1114 (2012)
7. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P.: Exploring strategies for training deep neural networks. The Journal of Machine Learning Research 10, 1–40 (2009)
8. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area v2. In: Advances in Neural Information Processing Systems, vol. 20, pp. 873–880 (2008)
9. Ma, J., Li, X.-D., Meng, Y.: Research of urban traffic flow forecasting based on neural network. Acta Electronica Sinica 37(5), 1092–1094 (2009)
10. Highway Capacity Manual. Highway capacity manual (2000)
11. Mohamed, A.-R., Dahl, G., Hinton, G.: Deep belief networks for phone recognition. In: NIPS Workshop on Deep Learning for Speech Recognition and Related Applications (2009)
12. Moorthy, C.K., Ratcliffe, B.G.: Short term traffic forecasting using time series methods. Transportation Planning and Technology 12(1), 45–56 (1988)
13. Salakhutdinov, R., Hinton, G.: Using deep belief nets to learn covariance kernels for gaussian processes. In: Advances in Neural Information Processing Systems, vol. 20, pp. 1249–1256 (2008)
14. Shuai, M., Xie, K., Pu, W., Song, G., Ma, X.: An online approach based on locally weighted learning for short-term traffic flow prediction. In: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 45. ACM (2008)
15. Smith, B.L., Demetsky, M.J.: Short-term traffic flow prediction: Neural network approach. Transportation Research Record (1453) (1994)
16. Sun, S., Xu, X.: Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems 12(2), 466–475 (2011)
17. Sun, S., Zhang, C.: The selective random subspace predictor for traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems 8(2), 367–373 (2007)
18. Sun, S., Zhang, C., Yu, G.: A bayesian network approach to traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems 7(1), 124–132 (2006)
19. Teh, Y.W., Hinton, G.E.: Rate-coded restricted boltzmann machines for face recognition. In: Advances in Neural Information Processing Systems, pp. 908–914 (2001)
20. Van Der Voort, M., Dougherty, M., Watson, S.: Combining kohonen maps with arima time series models to forecast traffic flow. Transportation Research Part C: Emerging Technologies 4(5), 307–318 (1996)
21. Yu, G., Hu, J., Zhang, C., Zhuang, L., Song, J.: Short-term traffic flow forecasting based on markov chain model. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 208–212. IEEE (2003)

Compact Prediction Tree: A Lossless Model for Accurate Sequence Prediction

Ted Gueniche[1], Philippe Fournier-Viger[1], and Vincent S. Tseng[2]

[1] Dept. of Computer Science, University of Moncton, Canada
[2] Dept. of Computer Science and Inf. Eng., National Cheng Kung University, Taiwan
{etg8697,philippe.fournier-viger}@umoncton.ca, tsengsm@mail.ncku.edu.tw

Abstract. Predicting the next item of a sequence over a finite alphabet has important applications in many domains. In this paper, we present a novel prediction model named CPT (*C*ompact *P*rediction *T*ree) which losslessly compress the training data so that all relevant information is available for each prediction. Our approach is incremental, offers a low time complexity for its training phase and is easily adaptable for different applications and contexts. We compared the performance of CPT with state of the art techniques, namely PPM (*P*rediction by *P*artial *M*atching), DG (*D*ependency *G*raph) and All-*K*-th-Order Markov. Results show that CPT yield higher accuracy on most datasets (up to 12% more than the second best approach), has better training time than DG and PPM, and is considerably smaller than All-*K*-th-Order Markov.

Keywords: sequence prediction, next item prediction, accuracy, compression.

1 Introduction

Given a set of training sequences, the problem of sequence prediction consists in finding the next element of a target sequence by only observing its previous items. The number of applications associated with this problem is extensive. It includes applications such as web page prefetching [3,5], consumer product recommendation, weather forecasting and stock market prediction.

The literature on this subject is extensive and there are many different approaches[6]. Two of the most popular are PPM (*P*rediction by *P*artial *M*atching)[2] and DG (*D*ependency *G*raph) [5]. Over the years, these models have been greatly improved in terms of time or memory efficiency [3,8] but their performance remain more or less the same in terms of prediction accuracy. Markov Chains are also widely used for sequence prediction. However, they assume that sequences are Markovian. Other approaches exist such as neural networks and association rules [9]. But all these approaches build prediction lossy models from training sequences. Therefore, they do not use all the information available in training sequences for making predictions.

In this paper, we propose a novel approach for sequence prediction that use the whole information from training sequences to perform predictions. The hypothesis is that it would increase prediction accuracy. There are however several

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 177–188, 2013.

important challenges to build such an approach. First, it requires a structure for storing the whole information efficiently in terms of storage space. Second, the structure should be efficiently updatable if new sequences are added. Third, it is necessary to define an algorithm for performing predictions using the data structure that is time efficient and generate accurate predictions.

We address all these challenges. First, we propose an efficient trie-based data structure named CPT (*C*ompact *P*rediction *T*ree) which losslessly compress all training sequences. The construction process of the CPT structure is incremental, offers a low time complexity and is reversible (i.e. it is possible to restore the original dataset from a CPT). Second, we propose an efficient algorithm to perform sequence predictions using the CPT structure. Thanks to CPT's indexing mechanism, the algorithm can quickly collect relevant information for making a prediction. Third, we introduce two strategies that respectively reduce the size of CPT and increase prediction accuracy. Lastly, we perform an extensive experimental study to compare the performance of our approach with state of the art sequence prediction algorithms, namely PPM [2](*P*rediction by *P*artial *M*atching), DG [5] (*D*ependency *G*raph) and All-*K*th-Order Markov [8], on several real-life datasets. Results show that CPT yield superior accuracy in most cases.

This paper is organized as follows. In section 2, we formally present the prediction problem and discuss related work. In section 3, we present CPT, explain how its substructures are built and how it is used to perform predictions. In section 4, we describe an experimental study. Finally, in section 5, we present our conclusions.

2 Preliminaries and Related Work

Problem Definition. Given a finite alphabet $I = \{i_1, i_2, ..., i_m\}$, an individual sequence is defined as $S = \langle s_1, s_2, ..., s_n \rangle$, a list of ordered items where $s_i \in I$ $(1 \leq i \leq m)$. Let $T = \{s_1, s_2, ..., s_t\}$ be a set of training sequences used to build a prediction model M. The problem of sequence prediction consists in predicting the next item s_{n+1} of a given sequence $\langle s_1, s_2, ..., s_n \rangle$ by using the prediction model M.

Related Work. *Prediction by Partial Matching* [2] (PPM) makes predictions based on the last K items of a sequence, where K defines the order of the model. A PPM model can be represented as a graph where prefix subsequences are linked to suffix subsequences by outgoing arcs having transition probabilities. In a K-Order PPM, the suffix of a given sequence is predicted by matching its last k items with one of the node. This approach has been proven to yield good results in certain areas [2,3]. However, an important drawback is its rigidness toward patterns that it can learn. The smallest variation in a subsequence will affect the prediction outcome, and thus prediction accuracy. This problem become worse for noisy datasets. In a K-Order PPM model only the Kth-order Markov predictor is used. In the *All-K-Order Markov Model* [8], all Markov predictors from 1 to K inclusively are used. This has the advantage of yielding higher

accuracy in most case [3]. But it suffers from a much higher state and space complexity. A lot of research has been done to improve the speed and memory requirement of these approaches, for example by pruning states [3,8,6].

The *Dependency Graph* (DG) [5] model is a graph where each node represents an item $i \in I$. A directionnal arc connects a node A to a node B if and only if B appears within x items from A in training sequences, where x is the *lookahead window length*. The weight of the arc is $P(B|A)/P(A)$.

There are many other approaches to sequence prediction such as using sequential rules [4], neural networks and Context Tree Weighting [10] (see [9] for an overview). However, all these approaches build lossy models, which may thus ignore relevant information from training sequences when making predictions. In this work, we propose a lossless prediction model. Our hypothesis is that using all the relevant information from training sequences to make predictions would increase prediction accuracy.

3 The Compact Prediction Tree

In this section, we present our approach. It consists of two phases: training and prediction.

3.1 Training

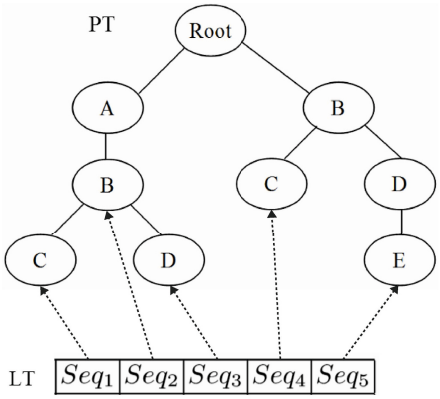
In the training phase, our prediction model named the Compact Prediction Tree is built. It is composed of three data structures: (1) a *P*rediction *T*ree (PT), (2) an *I*nverted *I*ndex (II) and (3) a *L*ookup *T*able (LT). The training is done using a training dataset composed of a set of sequences. Sequences are inserted one at a time in the PT and the II. For example, Figure 1 show the PT, II and LT constructed from sequences $\langle A, B, C\rangle$, $\langle A, B\rangle$, $\langle A, B, D\rangle$, $\langle B, C\rangle$, $\langle B, D, E\rangle$.

The *Prediction Tree* is recursively defined as a node. A node contains an item, a list of children nodes and a pointer to its parent node. A sequence is represented within the tree as a full branch or a partial branch; starting from a direct child of the root node. The prediction tree is constructed as follows: given a training sequence, we check if the current node (the root) has a direct child matching the first item of this sequence. If it does not, a new child is inserted to the root node with this item's value. Then, the cursor is moved to the newly created child and this process is repeated for the next item in the training sequence. The construction of this tree for N training sequences takes $O(N)$ in time and is done by reading the sequences one by one with a single pass over the data. The space complexity of the PT is in the worst case $O(N * averageLengthOfSequences)$ but in the average case the PT is more compact because the branches often overlap by sharing nodes. Two sequences share their first v nodes in the PT if they share a prefix of v items. The PT is incrementally updatable and is fast to construct.

The second structure is the *Inverted Index*. It is designed to quickly find in which sequences a given item appears. Hence, it can also be used to find all the

sequences containing a set of items. The II is defined as a hash table containing a key for each unique item encountered during the training. Each key leads to a bitset that indicates IDs of the sequences where the item appears. A bitset contains n bits, where n is the number of training sequences. The presence of an item in the s-th sequence is indicated by setting the s-th bit to 1 in its bitset, and 0 otherwise. The II, just like the PT, has an average construction time of $O(n)$ and takes $((n + b) * u)$ bytes where n is the number of training sequences, u is the number of unique items and b is the size of an item in bytes.

The third and last structure is the *Lookup Table*. It links the II to the PT. For each sequence ID, the LT points to the last node of the sequence in the PT. The LT purpose is to provide an efficient way to retrieve sequences from the PT using their sequence IDs. The LT is updated after each sequence insertion in the PT. Its time complexity is $O(n)$ where n is the number of sequences. In terms of size, this data structure takes $n * (b + p)$ bytes where n is the number of sequences, b is the size of an item in bytes and p is the size of a pointer in bytes. The addition of the LT to the PT makes it a lossless representation of the training set of sequences, i.e. it allows restoring the original dataset.



II

Item	Seq_1	Seq_2	Seq_3	Seq_4	Seq_5
A	1	1	1	0	0
B	1	1	1	1	1
C	1	0	0	1	0
D	0	0	1	0	1
E	0	0	0	0	1

Fig. 1. A Prediction Tree (PT), Inverted Index (II) and Lookup Table (LT)

The training process is really fast ($O(n)$). The CPT take more or less space depending on the dataset. If many sequences share common prefixes, a greater compression is achieved. Note that the PTitself could be further compressed by replacing frequent subsequences by single nodes or pruning infrequent nodes. These optimizations are outside the scope of this paper and will be investigated in future work.

3.2 Prediction

In the prediction phase, our prediction model is used to perform predictions. Let x be an integer named the prefix length. Making a prediction for a given sequence S is done by finding all sequences that contains the last x items from S

in any order and in any position. We call these sequences the *sequences similar to* S and they are used to predict the next item of S. The process of finding the sequences similar to S is implemented efficiently by using the II. It is done by performing the intersection of the bitsets of the last x items from S. The resulting bitset indicates the set of sequences similar to S. Using the LT, it is trivial to access these sequences in the PT. For each similar sequence Y, the algorithm capture its consequent w.r.t S. The *consequent of a sequence* Y *with respect to a sequence* S is the subsequence of Y starting after the last item in common with S until the end of Y. Each item of each of those consequents are then stored in a structure named *Count Table* (CT). A CT is defined as a hash table with items as keys and a score as associated value. This structure holds a list of possible candidate items and their respective score for a specific prediction and hence is unique for each individual prediction task. The item with the highest score within the CT is the predicted item. The primary scoring measure is the support. But in the case where the support of two items is equal, the confidence is used. We define the *support of an item* s_i as the number of times s_i appears in sequences similar to S, where S is the sequence to predict. The *confidence of an item* s_i is defined as the support of s_i divided by the total number of training sequences that contain s_i (the cardinality of the bitset of s_i in the II). We picked the support as our main scoring measure because it outperformed other measures in terms of accuracy in our experiments.

Performing a prediction is fairly fast. The time complexity is calculated as follows. The search for similar sequences is performed by bitset intersections (the bitwise AND operation), which is $O(1)$. The construction of the CT is $O(n)$ where n is the number of items in all consequents. Finally, choosing the best scoring item is done in $O(m)$ where m is the number of unique items in all consequents. In terms of spatial complexity, the CT is the only constructed structure in the prediction process and its hashtable only has m keys.

3.3 Optimizations

Sequence Splitter. The first optimization is done during the training phase while the PT is being constructed. Let *splitLength* be the maximum allowed length for a sequence. For each sequence longer than *splitLength* items, only the subsequence formed by its last *splitLength* items are inserted in the PT. By using this optimization, the resulting CPT is no longer lossless since sequence information is discarded. Splitting long sequences has for goal to reduce the PT size by reducing the number of possible branches and by enforcing an upper bound on the depth of branches. Intuitively, it may seems that this optimization would negatively affect the prediction's accuracy. But we have observed that it boosts the accuracy by forcing prediction to focus on the latest W items of each training sequence. We also observed that this optimization greatly reduces the prediction time and the CPT size (cf. section 4.3).

Recursive Divider. One of the problem we experienced early in our research is the low coverage of our approach for prediction. Since our model is based on

finding similar sequences that share a fixed subset of items T, if some noise is introduced in T, CPT is only able to find similar sequences containing the same noise. To make our approach more flexible, we introduce a recursive method named the Recursive Divider that tries removing the noise from T when searching for similar sequences. This approach works by levels $k = 1, 2...$ *maxLevel*, where *maxLevel* is a constant indicating the maximum number of levels to explore. At level k, for each subset $Q \subset T$ such that $|Q| = k$, the Recursive Divider uses the similar sequences to T/Q to update the CT. Note that each training sequence is only used once for each level to update the CT. If a prediction cannot be made at level k, the Recursive Divider moves to level $k+1$ if $k+1 < maxLevel$. In the experimentation section, we show that this technique boosts the coverage of CPT.

4 Experimental Evaluation

To evaluate the performance of the proposed prediction model, we performed a set of experiments. Our test environment is made of an Intel i5 third generation processor with 4.5 GB of available RAM on a 64-bit version of Windows8.

4.1 Datasets

We used five real-life datasets representing various types of data. Table 1 summarizes their characteristics. For each dataset, sequences containing less than 3 items were discarded .

BMS is a popular dataset in the field of association rule mining made available for KDD CUP 2000 [11]. It contains web sessions from an e-commerce website, encoded as sequences of integers, representing web pages.

FIFA contains web sessions recorded on the 1998 FIFA World Cup Web site and holds 1,352,804,07 web page requests [1]. Originally, the dataset is a set of individual requests containing metadata (e.g. client id and time). We converted requests into sequences by grouping requests by users and splitting a sequence if there was a delay of more than an hour between two requests. Our final dataset is a random sample from the original dataset.

SIGN is a dense dataset with long sequences, containing 730 sequences of sign-language utterances transcripted from videos [7].

KOSARAK is a dataset containing web sessions from a Hungarian news portal available at `http://fimi.ua.ac.be/data`. It is the largest dataset used in our experimental evaluation.

BIBLE is the religious Christian set of books used in plain text as a flow of sentences. The prediction task consists in predicting the next character in a given sequence of characters. The book is split in sentences where each sentence is a sequence. This dataset is interesting since it has a small alphabet with only 75 distinct characters and it is based on natural language.

Table 1. Dataset characteristics

Dataset	Sequence count	Unique items	Avg sequence length	Avg item occurence count per sequence
BMS	15,806	495	6.01	1.00
FIFA	28,978	3,301	32.11	1.04
SIGN	730	267	93.00	1.79
KOSARAK	638,811	39,998	11.64	1.00
BIBLE	32,529	76	130.96	4.78

4.2 Evaluation Framework

We designed a framework to compare our approach with state-of-the-art approaches on all these datasets. The framework is publicly available at `http://goo.gl/hDtdt` and is developed in Java. The following paragraphs describes the evaluation process of our framework.

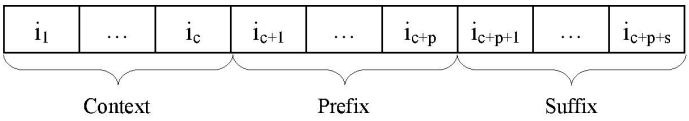


Fig. 2. Sequence splitting (context, prefix, suffix)

Each dataset is read in memory. Sequences containing less than three items are discarded. The dataset is then split into a training set and a testing set, using the 10-fold cross-validation technique. For each fold, the training set is used to train each predictor. Once the predictors have been trained, each sequence of the testing set is split into three parts; the *context*, the *prefix* and the *suffix* as shown in Fig. 2. The prefix and suffix size are determined by two parameters named *PrefixSize* (p) and *SuffixSize* (s). The context (c) is the remaining part of the sequence and is discarded. For each test sequence, each predictor accepts the prefix as input and makes a prediction. A prediction has three possible outcomes. The prediction is a *success* if the generated candidate appears in the suffix of the test sequence. The prediction is a *no match* if the predictor is unable to perform a prediction. Otherwise it is a *failure*. We define three measures to assess a predictor overall performance. *Local Accuracy* (eq. 1) is the ratio of successful predictions against the number of failed predictions.

$$Local\_Accuracy = |successes|/(|successes| + |failures|) \quad (1)$$

Coverage (eq. 2) is the ratio of sequence without prediction against the total number of test sequences.

$$Coverage = |no\_matches|/|sequences| \quad (2)$$

Accuracy (eq. 3) is our main measure to evaluates the accuracy of a given predictor. It is the number of successful prediction against the total number of test sequences.

$$Accuracy = |successes|/|sequences| \quad (3)$$

The above measures are used in our experiments as well as the spatial size (in nodes), the training time (in seconds) and the testing time (in seconds). The spatial size is calculated in nodes because the spatial complexity of all predictors can be represented in terms of nodes. This measure is meant to show the spatial complexity and is not used to determine the exact size of a model.

4.3 Experiments

Overall Performance. The goal of the first experiment consists in getting an overview of the performances (accuracy, training and testing time and space) of CPT against DG, 1st order PPM and All-Kth-Order Markov (AKOM). DG and AKOM were respectively tuned with a lookahead window of 4 and with an order of 5, since these values gave the best performance and are typically good values for these algorithms [3,5,8]. Results are shown in Tables 2 and 3. Results show that CPT yield a higher accuracy for all but one dataset. DG and PPM perform well in some situations but CPT is more consistent across all datasets. The training time, just like the testing time can be critical for some applications. In this experiment, CPT is always faster to train than DG and All-Kth-Order Markov by at least a factor of 3, and has comparable training time to PPM. The downside of CPT is that making a prediction can take longer than other methods. This characteristic is a trade off for the higher accuracy and is mainly caused by the Recursive Divider optimization described in section 3.3. The coverage is not presented because a high coverage ($> 95\%$) is achieved by all the predictor for all datasets and it is also indirectly included in the *overall accuracy* measure.

Table 2. Comparison of accuracy and model size

Dataset	Overall Accuracy				Size (nodes)			
	DG	CPT	PPM	AKOM	DG	CPT	PPM	AKOM
BMS	36.07	**38.45**	31.12	30.81	**484**	30920	**484**	67378
FIFA	25.87	**37.2**	24.44	27.98	**3027**	167935	**3027**	1397238
SIGN	3.54	**34.795**	4.11	10.14	**262**	4477	**262**	180396
KOSARAK	31.44	**34.26**	25.3	21.34	**16646**	234301	**16646**	1146462
BIBLE	6.26	82.06	29.06	**82.48**	**75**	11070	**75**	79456

Scalability. Our second experiment compares the scalability of each approach. The importance of scalability is application specific. But it is an important factor for most prediction tasks since the ability to scale of a prediction model can

Table 3. Comparison of training time and testing time

Dataset	Training time (s)				Testing time (s)			
	DG	CPT	PPM	AKOM	DG	CPT	PPM	AKOM
BMS	0.076	0.018	**0.01**	0.356	0.004	0.352	**0.001**	0.004
FIFA	3.032	0.153	**0.095**	12.347	0.301	0.146	**0.006**	0.085
SIGN	0.172	**0.008**	0.009	0.455	0.002	0.134	**0.001**	0.002
KOSARAK	9.697	0.741	**0.173**	6.051	0.042	1.533	0.018	**0.011**
BIBLE	0.803	**0.007**	0.244	4.031	0.018	0.029	0.043	**0.002**

directly or indirectly limit its accuracy and coverage. For this experiment, we used the FIFA dataset because of its high number of sequences and unique items. The experiment is conducted in steps, where the predictors are trained and tested with a higher number of sequences at each following step. Figure 3 shows the results in terms of accuracy, space and time. All training times in this experiment follow a linear evolution, but CPT and PPM operate at a much lower scale and are very close. PPM and DG have low spatial complexity because of their compact representation compared to CPT which takes more place but still grows linearly.

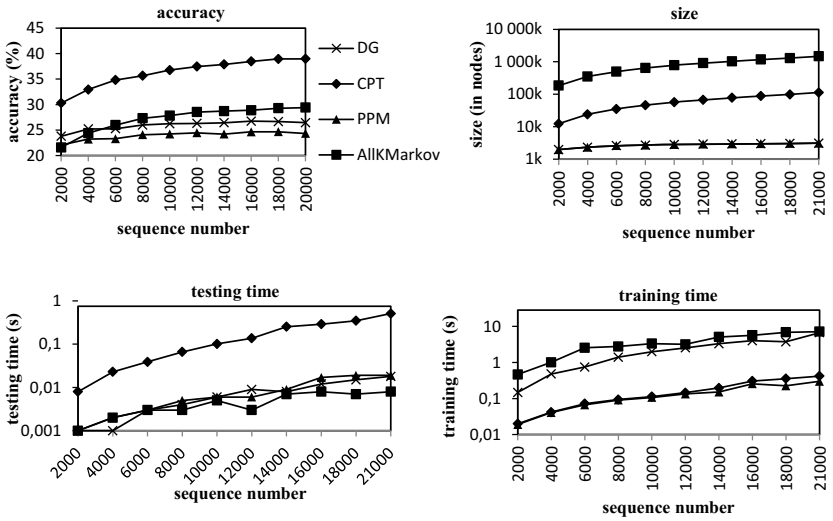


Fig. 3. Comparion of scalability

Prefix Length. The third experiment assesses the effect of the prefix size on the accuracy and coverage. Results are shown in Figure 4 for the FIFA dataset. Recall that the predictors output a prediction based on the prefix given as input. The longer the prefix, the more contextual information is given to the predictor.

Note that DG, PPM and All-Kth-Order Markov use a predetermined portion of the prefix defined by the order of each algorithm. Thus, by increasing the prefix length, we can observe that none of these algorithms get an increase in any performance measures. CPT takes advantage of a longer prefix by finding more precise (longer) patterns in its prediction tree, to yield a higher accuracy. The accuracy of CPT gets higher as the prefix length is raised. But after the prefix reaches a length of 8 (specific to the dataset), the accuracy decreases. This is because the algorithm may not be able to match a given prefix to any branches in the prediction tree. It means that this parameter should be finely tuned for each dataset to maximize the accuracy. The figure on the right of Fig. 4 shows the influence of the prefix length on the CPT spatial complexity.

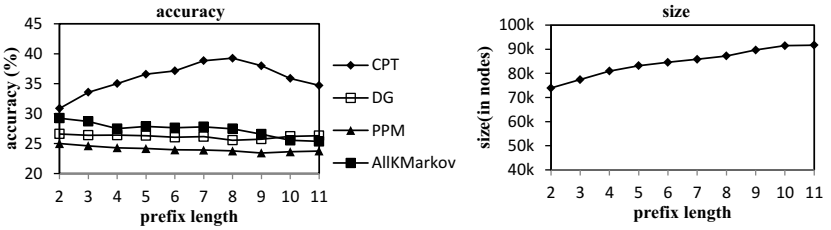


Fig. 4. Influence of prefix length on accuracy and model size

Optimizations. The fourth experiment assesses the influence of the Recursive Divider optimization (cf. Section 3.3) for CPT. The Recursive Divider aims at boosting the coverage of predictions using the CPT by ignoring items that could be noise during the prediction process. But it also indirectly influence accuracy. Figure 5 shows the effect of the Recursive Divider on the FIFA dataset by sequentially incrementing the *maxLevel* parameter. We can observe that the accuracy and the coverage of CPT are getting higher as the *maxLevel* parameter is raised. Also, the coverage and the accuracy measures quickly stabilize without affecting the testing time. This strategy's parameter can therefore be set to a really high value to guarantee the best coverage and accuracy and it does not need to be adjusted for each dataset.

The fifth experiment measures the influence of the Sequence Splitter optimization (cf. Section 3.3). It truncates long sequences before they are inserted in the prediction tree during the training phase. This makes the prediction tree more compact by reducing the number of possible branches and their depth. Reducing the depth improves the time complexity for both the training and testing processes. In this experiment we used the FIFA dataset because it has long sequences. We evaluated the performance of our model against different values for the *splitLength* parameter. For low values (eg. 5) most of the training sequences are split. By setting *splitLength* to a high value (eg. 40 or more), only a small number of sequences are splitted. In Figure 6, we show the effect of applying the Sequence Splitter strategy on the accuracy, the spatial size and the testing time,

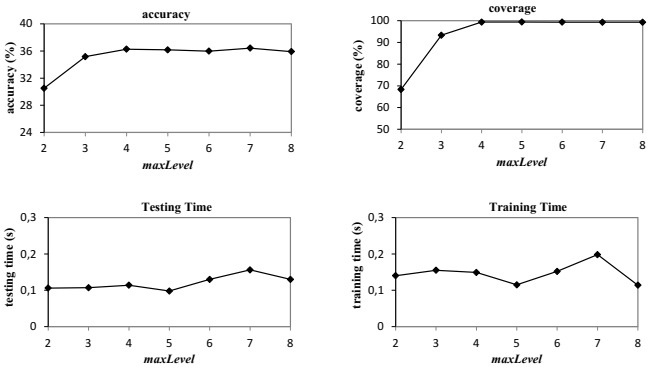


Fig. 5. Influence of the Recursive Divider optimization

for various split lengths. By setting *splitLength* to a low value (left side of each chart of Fig. 6), the spatial size is reduced by a factor of 7 while having a really low training and testing time and still having a high accuracy. Once again, this parameter should be finely tuned for each dataset if one wants to achieve the best performances.

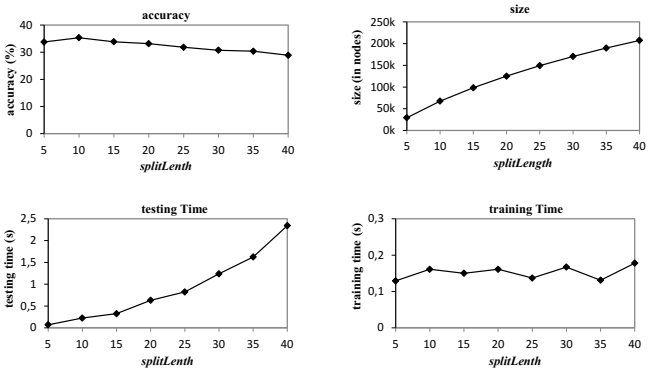


Fig. 6. Influence of the Sequence Splitter optimization

5 Conclusion

Predicting the next item of a sequence over a finite alphabet is essential to a wide range of applications in many domains. In this paper we presented a novel prediction model named the Compact Prediction Tree for sequence prediction. CPT is lossless (it can use all the information from training sequences to make a prediction), is built ly with a low time complexity. We also presented two optimizations (Recursive Divider and Sequence Splitter), which respectively boost the coverage of CPT and reduce its size.

We compared CPT to state-of-the-art approaches, namely PPM, All-Kth-Order Markov Model and DG on six real-life datasets. The source code of algorithms and datasets used in the experiments are available at `http://goo.gl/hDtdt`. Results show that CPT achieves the highest accuracy on all but one dataset with an accuracy up to 12% higher than the second best approach. CPT also shows better training time than DG and All-Kth Order Markov Model by at least a factor of 3. CPT is also considerably smaller than the All-Kth Order Markov model by at least a factor of 2. CPT is easily adaptable for different applications and contexts as shown in the experiments.

In the future, we aim to further improve the accuracy of CPT and its compression. We believe that higher compression can be achieved by grouping patterns of nodes and pruning nodes in the prediction tree. We also plan to compare our model against other prediction techniques such as Context Tree Weighting and Neural Networks.

References

1. Arlitt, M., Jin, T.: A workload characterization study of the 1998 world cup web site. IEEE Network 14(3), 30–37 (2000)
2. Cleary, J., Witten, I.: Data compression using adaptive coding and partial string matching. IEEE Trans. on Inform. Theory 24(4), 413–421 (1984)
3. Deshpande, M., Karypis, G.: Selective Markov models for predicting Web page accesses. ACM Transactions on Internet Technology 4(2), 163–184 (2004)
4. Fournier-Viger, P., Gueniche, T., Tseng, V.S.: Using Partially-Ordered Sequential Rules to Generate More Accurate Sequence Prediction. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS (LNAI), vol. 7713, pp. 431–442. Springer, Heidelberg (2012)
5. Padmanabhan, V.N., Mogul, J.C.: Using Prefetching to Improve World Wide Web Latency. Computer Communications 16, 358–368 (1998)
6. Domenech, J., de la Ossa, B., Sahuquillo, J., Gil, J.A., Pont, A.: A taxonomy of web prediction algorithms. Expert Systems with Applications (9) (2012)
7. Papapetrou, P., Kollios, G., Sclaroff, S., Gunopulos, D.: Discovering Frequent Arrangements of Temporal Intervals. In: Proc. of the 5th IEEE International Conference on Data Mining, pp. 354–361 (2005)
8. Pitkow, J., Pirolli, P.: Mining longest repeating subsequence to predict world wide web surfing. In: Proc. 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, CO, pp. 13–25 (1999)
9. Sun, R., Giles, C.L.: Sequence Learning: From Recognition and Prediction to Sequential Decision Making. IEEE Intelligent Systems 16(4), 67–70 (2001)
10. Willems, F., Shtarkov, Y., Tjalkens, T.: The context-tree weighting method: Basic properties. IEEE Trans. on Information Theory 31(3), 653–664 (1995)
11. Zheng, Z., Kohavi, R., Mason, L.: Real world performance of association rule algorithms. In: Proc. 7th ACM Intern. Conf. on KDD, pp. 401–406 (2001)

Generalization of Malaria Incidence Prediction Models by Correcting Sample Selection Bias

Orlando P. Zacarias[1,2,⋆] and Henrik Boström[1]

[1] Department of Computer and Systems Sciences
Stockholm University
Forum 100, SE-164 40 Kista, Sweden
[2] Department of Mathematics and Informatics - Faculty of Science
Eduardo Mondlane University
Main Campus, P.O. Box 250, Maputo, Mozambique
{si-opz,henrik.bostrom}@dsv.su.se, ozacas@uem.mz

Abstract. Performance measurements obtained from dividing a single sample into training and test sets, e.g. by employing cross-validation, may not give an accurate picture of the performance of any model developed from the sample, on the set of examples to which the model will be applied. Such measurements, which are due to that training and test samples are drawn according to different distributions may hence be misleading. In this study, two support vector machine models for predicting malaria incidence developed from certain regions and time periods in Mozambique are evaluated on data from novel regions and time periods, and the use of selection bias correction is investigated. It is observed that significant reductions in the predicted error can be obtained using the latter approach, strongly suggesting that techniques of this kind should be employed if test data can be expected to be drawn from some other distribution than what is the origin of the training data.

Keywords: prediction, generalization, sample selection bias, malaria incidence.

1 Introduction

The primary health system in Mozambique is the main health-care provider for the majority of the population. With a scarce health infrastructure, it takes more than an hour for most patients to walk to the nearest health center [1]; particularly in rural areas. The situation is exacerbated with health units facing frequent disruptions of drug stocks and a general lack of basic services. Furthermore, human resources for health care are severely constrained and often poorly trained with a limited management skills. All this posses a great challenge to the health authorities in Mozambique. Moreover, the available health information and monitoring system is generally unable to provide timely and accurate health prediction information.

⋆ Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 189–200, 2013.

The provision of accurate malaria incidence prediction is possibly one of the most important and efficient ways by which health authorities and other related stakeholders can monitor the disease and plan their everyday activities towards providing of better health services and interventions to the citizens and communities [2]. The dissemination of obtained predictions by the health authorities may promote open access to disease and health information to citizens, improve government efficiency and service delivery, strengthen accountability and increase transparency in the management of their activities [1]. Access to evidence-based malaria services may improve the quality of the health care system in Mozambique, develop better health-seeking specific behaviors through awareness raising, and ultimately contribute to mortality and malaria incidence reduction.

Classification and prediction based on *Support Vector Machines (SVMs)* has been widely employed [3,4,5]. Through this and other well known data mining techniques, it is possible to extract relevant information in the form of predictive models from case based health data [6]. In fact, the development of prediction models within the health sector is meant to guide health professionals in their decision-making process as for instance, to improve management. Hence, in [7,8], we investigated the application of SVMs to construct two malaria prediction models based on the *support vector regression* technique. The predictive performance was evaluated using two hold-out sets; one for infants and the other considering malaria data from all age groups. Both models were derived for the same region of the province of Maputo in Mozambique, due to the availability of data in consecutive years in the region. The first model was developed using malaria data cases for the years 2007-2008 from infants (0-4 years of age), whereas the second model was built from malaria cases covering all ages for the period of nine consecutive years (1999-2007).

When executing a data mining project, the validation of derived models is crucial. Actually, application of internal validation techniques such as cross-validation is not often sufficient [9]. In most cases, when applied to new samples, the performance of predictive models is generally lower than what is observed when making predictions using the original sample, even when excluding training examples. Thus, when the developed models have passed the internal validation process, they should undergo further testing or evaluation on new test data before being applied in practice. This is an important step, especially for data mining studies within the health sector [9,10,11,12]. For the particular case of malaria, the situation is striking given that conditions such as mosquito (vector) population density, environmental climatic factors, population characteristics, etc., vary from one region to another. Therefore, to evaluate the capabilities of the models to generalize to any external dataset, they are in this study tested for applicability and reliability [13] using two new datasets obtained from other regions (provinces) of the country. Moreover, these datasets were collected from regional (administrative) health centers located in different provinces and sampled at different (future) time points, from where the prediction models were developed. Actually, the more these new testing datasets differ from the datasets used in the

model development, the stronger the test of generalization of the model becomes [9]. Hence in this study, we compare the predictive performance on these new datasets to the accuracy obtained from cross-validation on the original datasets [7,8] to assess the reliability and usefulness of the derived malaria incidence cases prediction models.

The application of these testing procedures become crucial, considering the goal of delivering an important and accessible means of malaria disease predictions. Literally, this can be taken as part of the evaluation phase, i.e. the step just before the knowledge deployment phase of the data mining process development [14]. Moreover, it is a step prior to integration of the data models within the Mozambican health information system; towards closing the cycle of this data mining project [11].

Datasets from year 2011 were selected, which was primarily motivated by their completeness and that they show similar structure as the data employed to develop the prediction models. Consequently, the data sets used in this generalization testing process are not chosen at random. This leads to the occurrence of the well known problem of *sample selection bias (SSB)* [15,16,17], where the learning model is developed from a training set that is sampled using a different distribution than what is used for obtaining the test set. Such a situation violates the traditional assumption of machine learning, which presupposes the same distributions for the training and test datasets, which normally leads to poor performance of the resulting predictive model [16]. For practical problems such as malaria incidence prediction, it is most likely that this assumption is violated [15,16]. Therefore, we evaluate the use of a strategy for correcting the sample selection bias [17] in both models. Mean square error, which is defined in the usual way as the sum of squared differences of predicted and actual malaria cases, divided by the number of actual data points in the study area, is used as the performance metric in this study.

The remainder of this article is organized as follows: in section two, we present the background to this study including data collection, processing and analysis, and a brief overview of the validation framework. Section three presents and discusses the empirical results. Finally, conclusions and future research work are outlined in section four.

2 Methods

This section is divided into four parts: first, a short description of SVMs is provided, followed by a presentation of the processes for data collection and analysis. Then the employed method for sample bias correction is described, and finally, the validation procedure is discussed.

2.1 SVMs

Support-vector machines (SVMs) are supervised learning models that were originally devised for classification problems [18]. As such, they determine decision

functions in the form of hyperplanes [19], and the corresponding learning algorithms are searching for hyperplanes with maximum margin. The method has been extended to address regression tasks with the algorithm exploring the problem space to optimize a cost function. In the linear case, the solution to a support vector problem is given by a linear combination of points lying in the decision surface of the hyperplane, within the margin of the classifier. These points are known as *support vectors* [20]. When no linear separating hyperplane can be found in the original feature space, the *kernel trick* is employed to efficiently project the original space into a new, typically much larger, space in which a linear separating hyperplane may be found [18]. Further discussions of methodological issues and parameter settings of SVMs that also are employed in this study are described elsewhere [7,8].

2.2 Data Collection, Processing and Analyses

Given the need for generalization of the prediction models of malaria incidence to other provinces different from the regions where these models were derived, we investigate the models developed in [7] and [8] by assessing their reliability and usefulness in correctly predict malaria incidence cases in these new regions. The data was obtained from the *Ministry of Health (MoH)*, namely the number of malaria cases and indoor-residual spray activities, while the *National Institute of Meteorology (INAM)* provided the climatic factors. The dataset covered the period of twelve years from 2000 to 2012. However, the health data was yearly aggregated for the period 2009-2010 and thus not suitable for use in this study, whilst the climatic data showed a high rate of missing values in 2012. This left us with the data from year 2011 as the only available option for performing the current study. We choose for the analysis two different provinces, the Zambézia province, which is located in the center of Mozambique and Cabo Delgado, which is situated in the north of the country. While the models 1 and 2 below were developed using datasets from the southmost province of the country, i.e., the Maputo province. The data was monthly aggregated per district in both the provinces of Cabo Delgado and Zambézia. Nine attributes were used: administrative districts and month of the year (categorical variables); number of malaria cases, temperature, precipitation, humidity and indoor-residual spray (numeric variables). The temperature attribute was further used to derive the attributes *minimal* and *maximal temperature*, as well as *temperature variation*, which is the difference of the two former attributes. This avoided the use of the *average temperature* since this quantity is considered as containing less information compared to the *temperature variation* for the development of malaria vectors and their survival [21].

Evaluation was performed using the two previously developed models:
Model 1 - Infants [7]: This model was built using data of malaria cases from a sample of the infants population (0-4 years of age) together with climatic factors and indoor-residual spray intervention, for the years 2007-2008. Model testing was initially performed using data for year 2009 in the Maputo province.

Model 2 - All ages [8]: In this case, the derivation of the model followed a time-frame approach using a block of nine consecutive years corresponding to the period 1999-2007. The same variables were considered as for the previous model. The model was first tested on data for 2008. Both initial tests for generalization apply a temporal [9] approach, i.e., the testing data sample are from the same region but only from later (future) time period.

Both the provinces of Cabo Delgado and Zambézia were used in the analysis employing the prediction model 2 (all ages) above, while only the data for the Zambézia province was suitable for use with model 1 (infants). The latter was mainly due to the fact that malaria cases for infants in province of Cabo Delgado were combined with the all ages malaria data, thus, not being possible to separate them. Districts of Quissanga and Namarroi in Cabo Delgado and Zambézia province respectively, were not included in the analysis because they show inconsistent data records. Additionally, each one of the provinces is subdivided into seventeen administrative districts. To keep the format of the derived prediction models 1 and 2, the districts in each province were randomly grouped into two series of eight districts. Table 1 shows the mean and standard deviation of the number of malaria cases in each province and district groups; with 28% of malaria cases affecting infants in Zambézia province. Moreover, the spread-out of raw malaria cases data is higher in districts of province of Zambézia 1 group followed by the set of districts in Cabo Delgado 1 in the entire population datasets. While in infants datasets, high spread of malaria cases is observed for Zambézia I group of districts.

Table 1. Statistics of malaria cases per province and district group

Cases from the entire population (all ages)		
Province and District Group #	Mean	Standard Deviation
Cabo Delgado 1	1471	1231.00
Cabo Delgado 2	1236	551.00
Zambézia 1	2517	1531.00
Zambézia 2	1666	892.00
Cases from infants (< 5 *years*)		
Zambézia I	622	650.00
Zambézia II	543	492.00

Initially, Microsoft Excel was used to process and analyze the data. Around 2.35% of climatic factors were missing. To meet the SVMs requirements, data preparation went through three main stages:

1. Replacement of missing attribute values. We use the average of all values specified for each attribute of the corresponding data set.
2. Resorting to the facilities provided in the Weka software [22] for data pre-processing, the data were prepared in the appropriate format for further pre-processing as follows:

(a) transformation of each numeric variable (number of malaria cases, climatic factors and indoor-residual spray activities) through min-max normalization.
(b) transformation of categorical variables (name of administrative district and month) to binary variables. In this case, each district and month becomes an attribute on its own.

3. Employment of Microsoft Excel to create appropriate *text input file* for use within the R-package [23] for validation purposes.

As a result, we obtained 27 attributes (variables) from the initial nine, namely: administrative districts (eight binary attributes), month of the year (twelve binary attributes), maximum, minimum and temperature variation, precipitation, humidity, indoor-residual spray and number of malaria cases. Hence, each datapoint is a combination of eight possible values of administrative districts (space domain), twelve values of different months (time domain) and a single occurrence (value) of the remaining attributes ($8 \times 12 \times 1 \times 1 \times 1 \times 1 \times 1 = 96$). A total of ninety six such combination is obtained for each district group dataset. The external sets of Zambézia I and II contain climatic, number of malaria cases and indoor-residual data for children less than five years of age and are used for testing the spatial and temporal generalization of infants developed **model 1** [7]. The spatial and temporal generalization ability of **model 2** [8] was evaluated on four datasets, i.e., from the district groups Cabo Delgado 1 and 2 and Zambézia 1 and 2 of Cabo Delgado and Zambézia provinces respectively.

2.3 Sample Selection Bias Correction for Malaria Prediction

The problem of correcting sample selection bias (SSB) was first studied by Heckman in [24]. Then, several studies [25,16,26] have introduced and investigated the *sample selection bias* problem or *covariate shift* [27] in the field of machine learning. However, a fundamental assumption of many learning schemes is based on the fact that the training and test data are independently drawn from an identical distribution. The violation of this (iid) assumption can lead to the sample selection bias problem. This problem may affect the learning of a classifier by reducing its prediction performance. Zadrozny [16] has recently introduced a solution to the SSB problem when the difference in the distributions of training and testing data arise due to non-random selection of examples. In machine learning, the SSB correction strategy consists mainly in re-weighting the cost errors associated with each training point of the biased sample as to reflect the unbiased distribution [15]. The correction is effective if we are able to explicitly allocate an identical distribution for both the training and test datasets.

In the current study, six new datasets are used to analyze the generalization ability of derived prediction models in Mozambique. The datasets are obtained from two different provinces and time periods from where the prediction models were originally developed. The provinces are non-randomly selected out of ten possible regions. Thus, the use of standard error estimation procedures such as cross-validation in the presence of sample selection bias, may result in poor

estimates of predictive performance when applied to the test samples. The aim is basically to choose a classifier/predictor that minimizes the expected prediction error on the test data. This will avoid the choice of a suboptimal model.

To get an unbiased estimate of the test error, we use importance weighting [17], where examples of the training set are weighted using an estimate of the ratio between test and training sets through feature densities. According to [17], the estimation of importance weighting on the training data may be given by,

$$\begin{aligned} w_i &= \frac{P_T(x_i, y_i)}{P_L(x_i, y_i)} \\ &= \frac{P_T(x_i)P_T(y_i|x_i)}{P_L(x_i)P_L(y_i|x_i)} \\ &= \frac{P_T(x_i)}{P_L(x_i)} \end{aligned} \tag{1}$$

where i is a training object with given feature vector x_i and response y_i, where $\mathbf{T}$ and $\mathbf{L}$ are training and test sets respectively (see [17] for further details). $P_T(.)$ and $P_L(.)$ are probability distributions of obtaining examples from training and test sets. The training set defined as $L = \{(x, y) \in X \times Y\}$, contains vectors from feature and response spaces, whereas test set contains feature vectors only. Given a feature vector, the probability of response is the same in both training and test sets and is represented by equation $P_T(y_i|x_i) = P_L(y_i|x_i)$, in second row of (1).

The estimation of training and test distributions was performed by employing kernel density estimation with a Gaussian kernel. Selection of kernel bandwidth was achieved by applying a data-driven strategy within the web-based optimization application [28]. This resulted in the bandwidth value of 0.02 for infants model and 0.04 for the model of all ages. To estimate the distributions $P_L(x_i)$ and $P_T(x_i)$, we adopted the kernel density estimation (KDE) procedure following [27], where a Gaussian kernel was employed. The learner is re-trained using the weighted values determined in equation (1) above as its training set, thereby obtaining a new prediction model as a result of the applied correction. Then, corrected prediction estimates are obtained based on new test sets.

The strategy of re-weighting the cost errors of each training point to correct the SSB can be applied with several classification and regression algorithms [17]. Mathematical details of the application of these techniques employing the *support vector machines* approach can be found in [17].

2.4 Validation Procedure

The performance of a predictive model should not be evaluated on the same dataset that was used for generating it. In fact, even the application of internal validation techniques, such as cross-validation, may not be sufficient [9], given the need for the models to be able to generalize to other samples, e.g., from the same region and future times, or even in other regions and times.

Thus, it is essential to test the prediction models of the malaria incidence to other provinces different from the regions where the models were derived, prior to their presentation and deployment to local health authorities. Generally this means to investigate the predictive performance of these models, thus assessing their reliability and usefulness in correctly predicting malaria incidence cases in new regions.

The applied spatial and temporal *external validation* procedure [29] is mainly expected to show the applicability of derived models 1 and 2, to predict malaria incidence cases in other provinces of Mozambique. Moreover, the application of external validation may lead to concluding that the models need to be revised in order to improve prediction of the number malaria cases in these other regions [13].

To implement the procedure for *external validation*, a program written in R [23] using the *e1071 package* [30], was employed to evaluate predictions of malaria incidence cases using the models 1 and 2 developed in [7,8]. As described above, for both models, datasets for 2011 from two different provinces were employed to perform the testing.

3 Results

We present the results divided into two parts: the first presents the results of using one validation dataset from the Zambézia province to which model 1 (infants) was applied. The second part presents the validation results of using four datasets sampled from the entire province population in provinces of Cabo Delgado in the north and Zambézia in the center part of the country, to which model 2 (all ages) was applied.

The framework of applying the generalization procedure on our prediction models by employing the kernel density estimation to correct the sample selection bias is compared to the standard approach that directly uses the learner to predict new incidence malaria cases, i.e, without performing sample selection correction to accurately predict malaria incidence. Evaluation of models was performed using new sets, by analyzing their predictive performance.

3.1 Testing Generalization of Model 1 (Infants)

Table 2 shows empirical results of the performed testing for generalization of the model 1 developed using support vector regression. It can be seen that the best predictive performance was obtained when applying the sample selection bias correction strategy as opposed to direct use of model 1.

We can also see from Table 2 that compared to the standard approach, a reduction of 33% in the error rate is achieved when testing the model on the Zambézia I group of districts using the prediction model with corrected sample selection bias. Similarly, the generalization testing after conducting sample selection bias correction gets a performance improvement in Zambézia II set with an error rate reduction of 15%.

Table 2. Mean Error of Predicted Malaria Cases of Infants Model

No Sample Selection Bias Correction	
Test Set Used	Mean Squared Error
Zambézia I	0.073107
Zambézia II	0.148522
With Sample Selection Bias Correction	
Zambézia I	0.04882762
Zambézia II	0.1264137

3.2 Generalization Test of Model 2 (all ages)

The model developed for all ages, was tested for generalization of its predictive performance on four different datasets of the year 2011, in two different provinces - Cabo Delgado in the north and Zambézia in center of the country. The obtained experimental results are shown in Table 3.

Table 3. Mean Error of Predicted Malaria Cases from model 2

No Sample Selection Bias	
Test Set Used	Mean Squared Error
Zambézia 1	0.1354169
Zambézia 2	0.9632578
Cabo Delagado 1	0.4219064
Cabo Delagado 2	0.8186807
After Sample Selection Bias Correction	
Zambézia 1	0.1017059
Zambézia 2	0.9718783
Cabo Delagado 1	0.3878769
Cabo Delagado 2	0.8404788

Similarly to the above, results with lowest estimated mean square error are obtained after a carefully approximation of probability densities of training and test sets, i.e., following the application of sample selection bias correction procedure. An error reduction rate of 25% is achieved for the Zambézia 1 groups of districts after bias correction, whereas for the Zambézia 2 set, the error increases by 1.0%. For the Cabo Delgado province, an increased error rate of 3.0% is observed for the second group of districts. However, a reduction of 8.0% in error rate can be observed for the first group of districts after employing the sample selection bias correction framework.

4 Conclusion

In this study we have investigated the capabilities of two two support-vector machine models to generalize to external datasets, by evaluating their accuracy on predicting malaria incidence cases in regions of Mozambique that differ from the ones used for training. Due to absence of randomness in choosing datasets for training and testing, the problem of sample selection bias may be an issue with apparent differences in distribution densities between training and test sets. Correction of sample selection bias was attempted by employing the Gaussian kernel density estimation technique to automatically generate a corrected distribution of the training features adjusted to the distribution of test set.

To help determine the relevance and necessity of applying the approach of sample selection bias in the prediction of malaria incidence, the models were also analyzed with a strategy where we directly applied the learner to test for generalization of predictions. Our empirical study shows that the employment of sample selection bias approach can substantially improve the performance of malaria incidence predictions when the nonrandom (biased) data sampling problem is encountered, although the improvement is not guaranteed.

The adoption of these prediction estimates by local health authorities, may improve the management of the disease and health decision-making, the health of the Mozambican population and are likely to reduce real cost of health service providers.

This investigation could be extended in several different directions:

- considering other strategies to deal with the sampling selection bias problem, e.g., active learning or co-training.
- re-calibrate the developed models as to include attribute reduction or even extensions by adding predictors. The later is important because some factors related to the incidence of malaria were ignored as the data were not available.

Acknowledgment. The authors thank the Mozambique Ministry of Health and the National Institute of Meteorology for their support and provision of data to conduct this study. We also thank Sida/Sarec and Eduardo Mondlane University project - *Global Research in Mathematics, Statistics and Informatics* for funding this research.

References

1. The United States Global Health Initiative, Mozambique-national health strategy 2011-2015 (2012), `http://hingx.org:8080/svn/main/eHealth%20Regulation/`
2. World Health Organization, Using Climate to Predict Infectious Disease Outbreaks: A Review (2004), `http://www.who.int/globalchange/publications/en/`
3. Liao, L., Noble, W.S.: Combining pairwise sequence similarity and support vector machines for detecting protein evolutionary and structural relationships. Journal of Computational Biology 10(6), 867–868 (2003)

4. Han, L.Y., Cai, C.Z., Lo, S.L., Chung, M.C.M., Chen, Y.Z.: Prediction of RNA-biding proteins from primary sequence by support vector machine approach. RNA Journal 10, 355–368 (2004)
5. Byvatov, E., Schneider, G.: SVM-based feature selection for characterization of focused compound collections. Journal of Chemical and Information Computer Science 44(3), 993–999 (2004)
6. Viademonte, S., Burstein, F.: From knowledge discovery to computational intelligence: A framework for intelligent decision support systems. Series on Intelligent Decision-making Support Systems, pp. 57–78. Springer-Verlag, London Limited, London (2006)
7. Zacarias, O.P., Boström, H.: Strengthening the Health Information System in Mozambique through Malaria Incidence Prediction. In: Proceedings of IST-Africa International Conference, Nairobi, Kenya (May 2013)
8. Zacarias, O.P., Boström, H.: Comparing Support Vector Regression and Random Forests for Predicting Malaria Incidence in Mozambique. In: Proceedings of International Conference on Advances in ICT for Emerging Regions, Colombo, Sri-Lanka (to appear, December 2013)
9. Moons, K.G.M., Kengne, A.P., Grobbee, D.E., Royston, P., Vergouwe, Y., Altman, D.G., Woodward, M.: Risk prediction models: II. External validation, model updating, and impact assessment. Biomedical Heart Journal (March 2012), doi:10.1136/heartjnl-2011-301247
10. Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice Hall, Pearson Education, Inc., Upper Saddle River, New Jersey (2002)
11. Obenshain, M.K.: Application of Data Mining Techniques to Healthcare Data. Journal of Infection Control and Hospital Epidemiology 25(8), 690–695 (2004)
12. Temu, E.A., Coleman, M., Abilio, A.P., Kleinschmidt, I.: High Prevalence of Malaria in Zambezia, Mozambique: The Protective Effect of IRS versus Increased Risks Due to Pig-Keeping and House Construction. PLoS ONE 7 (2012), doi:10.1371/journal.pone.0031409
13. Steyerberg, E.W., Borsboom, G.J.J.M., van Houwelingen, H.C., Eijkemans, M.J.C., Habbema, J.D.F.: Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. Journal of Statistics in Medicine 23, 2567–2586 (2004)
14. Clifton, C., Thuraisingham, B.: Emerging standards for data mining. Journal of Computers Standards and Interfaces 23, 187–193 (2001)
15. Cortes, C., Mohri, M., Riley, M.D., Rostamizadeh, A.: Sample Selection Bias Correction Theory. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNCS (LNAI), vol. 5254, pp. 38–53. Springer, Heidelberg (2008)
16. Zadrozny, B.: Learning and Evaluating Classifiers under Sample Selection Bias. In: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada (2004)
17. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting Sample Selection Bias by Unlabeled Data. In: NIPS (2007)
18. Boser, B.E., Guyon, I., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, pp. 144–152. ACM Press, New York (1992)
19. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Inc., New York (2001)

20. Barbella, D., Benzaid, S., Christensen, J., Jackson, B., Qin, X.V., Musicant, D.: Understanding Support Vector Machine Classifications via a Recommender System-Like Approach. In: Proceedings of the International Conference on Data Mining (DMIN), Las Vegas, USA, July 13-16 (2009)
21. Blanford, J.I., Blanford, S., Crane, R.G., Mann, M.E., Paaijmans, K.P., Schreiber, K.V., Thomas, M.B.: Implications of temperature variation for malaria parasite development across Africa. Scientific Reports 3(1300) (2013), doi:10.1038/srep01300
22. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
23. R-Statistical tool for data analysis, `http://CRAN.R-project.org/` (accessed September 16, 2012)
24. Heckman, J.J.: Sample selection bias as a specification error. Econometrica 47(1), 153–161 (1979)
25. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI, pp. 973–978 (2001)
26. Fan, W., Davidsno, I., Zadrozny, B., Yu, P.S.: An improved categorization of classifier's sensitivity on sample selection bias. In: ICDM, pp. 605–608. IEEE Computer Society, Los Alamitos (2005)
27. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference 90, 227–244 (2000)
28. Shimazaki, H., Shinomoto, S.: Kernel bandwidth optimization in spike rate estimation. Journal of Computer Neuroscience 29, 171–182 (2010)
29. Hernandez, N., Kiralj, R., Ferreira, M.M.C., Talavera, I.: Critical comparative analysis, validation and interpretation of SVM and PLS regression models in a QSAR study on HIV-1 protease inhibitors. Journal of Chemometrics and Intelligent Laboratory Systems 98, 65–77 (2009)
30. Meyer, D., Dimitriadou, E., Hornik, K., Weingesse, A., Leisch, F.: Manual of Package e1071, `http://CRAN.R-project.org/=e1071` (accessed October 2012)

Protein Interaction Hot Spots Prediction Using LS-SVM within the Bayesian Interpretation

Juhong Qi[1,2], Xiaolong Zhang[1,2,*], and Bo Li[1,2]

[1] School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, 430065, China
[2] Hubei Key Lab of Intelligent Information Processing and Real-time Industrial System, Wuhan, 430065, China
qjhth001@gmail.com, xiaolong.zhang@wust.edu.cn, liberol@126.com

Abstract. Hot spot prediction in protein interfaces is very important for understanding the essence of protein interactions and may provide promising prospect for drug design. Since experimental approaches such as alanine scanning mutagenesis are cost-expensive and time-consuming, reliable computational methods are needed. In this paper, a systematic method based on least squares support vector machine (LS-SVM) within the Bayesian evidence framework is proposed, where three levels Bayesian inferences are used to determine the model parameters and regularization hyper-parameters. Then a higher precision model for hot spots is constructed by optimizing these parameters. Compared with the previous methods, our model appears to be better performance.

Keywords: Hot spot, Prediction, Protein interface, LS-SVM, Three levels Bayesian inferences.

1 Introduction

Understanding both the structure and the biological function of proteins, which is the elementary blocks of all living organisms, is a longstanding and fundamental topic in biology [1]. It will contribute to cure diseases with newly designed proteins with predefined functions to solving the problem.

The previous studies have discovered that proteins form certain active 3D structures can interact with other molecules through their interfaces [1]. These protein-protein interactions play a crucial role in signal transduction and meta bolic networks. It is also well known that the distribution of binding energies on the interface is not uniform [2]. Moreover, small critical residues termed as hot spots contribute a large fraction of the binding free energy, which are crucial for preserving protein functions and maintaining the stability of protein interactions. In the binding interface, hot spots

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 201–210, 2013.

are packed significantly more tightly than other residues. These hot spots are also surrounded by residues that are energetically less important [3]. Therefore, it is very important to identify hot spots in protein interfaces for understanding protein-protein interaction.

Because the physicochemical experimental method such as alanine scanning mutagenesis is time-consuming and labor-intensive and is only used for hot spots prediction in a limited number of complexes, there is an urgent need for computational methods to predict hot spots. In recent years, some computational methods have been proposed to predict hot spots. Tuncbag [4] established a web server HotPoint combining solvent accessibility and statistical pairwise residue potentials to predict hot spot computationally. Darnell [5] also provided a web server KFC to predict hot spots by decision tree with various features. Cho [6] developed two feature-based predictive SVM models with features such as weighted atom packing density, relative accessible surface area, weighted hydrophobicity and molecular interacion types. Xia [7] introduced an ensemble classifier based on protrusion index and solvent accessibility to boost hot spots prediction accuracy. Recently, Zhang [8] applied support vector machines (SVMs) to predict hot spots with features such as weighted residue contact, relative accessible surface area, accessible surface area, weighted hydrophobicity and protrusion index. Although these methods have predicted hot spots effectively, there are still some problems remaining in this area. First of all, effective feature selection methods and useful feature subsets have not been found yet. Moreover, it is evident that the limitations exist in the prediction performances.

LS-SVM is proposed by Suykens [9] based on SVM, which can reduce the computational complex by introducing equality constrains and least square error to obtain a linear set of equations in the dual space . But the traditional least squares support vector machine (LS-SVM) model, using cross validation to determine the regularization parameter and kernel parameter, is time-consuming. Bayesian framework is applied to infer the hyper-parameters used in LS-SVM so as to eliminate the work of cross-validation. The optimal parameters of LS-SVM are obtained by maximizing parameter distribution a posteriori probability since Bayesian framework is to maximize the parameter distribution a posteriori probability [9].

In this paper, we propose a new method for predicting hot spots in protein interfaces. Firstly, we extract features from protein sequence and structure information. Then we employ two-step method to remove noisy and redundant features. Then we apply LS-SVM in Bayesian inference to identify hot spots in protein-protein interface. In the end, we evaluate the method by 10-fold cross-validation and independent test. Compared with other previous methods, our model obtains better results.

2 Introduction of LS-SVM

LS-SVM is derived from the standard Vapnik SVM classifier by transforming the QP problem into a linear system problem as follows [9]:

$$\min_{w,b,\xi} J(w,\xi) = \frac{1}{2} w^T w + \frac{1}{2} c \sum_{i=1}^{l} \xi_i^2$$
$$s.t.: y_i = w^T \phi(x_i) + b + \xi_i, i = 1,2,\cdots,l \tag{1}$$

where $\phi(\cdot)$ is the nonlinear mapping function (kernel function), which maps samples into the feature space; w is the weight vector, b the bias term, ξ_i are error variables and c is a adjustable hyper-parameter.

Because of w in the feature space, it is difficult to solve w directly. Therefore, we will obtain a solution in the dual space. Then, the Lagrangian of the problem is expressed as

$$L(w,b,\xi,\alpha) = \frac{1}{2} w^T w + \frac{1}{2} c \sum_{i=1}^{l} \xi_i^2 - \sum_{i=1}^{l} \alpha_i \{ y_i - [w^T \phi(x) + b] - \xi_i \} \tag{2}$$

where α_i are Lagrange multipliers with either positive or negative value.

According to Karush-Kuhn-Tucker conditions, the solution can be obtained by solving the following linear equations.

$$\begin{pmatrix} K + C^{-1} I & \vec{1}^T \\ \vec{1} & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ b \end{pmatrix} = \begin{pmatrix} Y \\ 0 \end{pmatrix} \tag{3}$$

where $K_{ij} = K(x_i, x_j)$, $Y = (y_1, \cdots, y_l)^T$, $\alpha = (\alpha_1, \cdots, \alpha_l)^T$, $\vec{1} = (1, \cdots, 1)$, $I = diag(1, \cdots, 1)$ .Thus, we can obtain α and b by solving (3). The standard LS-SVM model can be drawn as follows:

$$y(\mathrm{x}) = \sum_{i=1}^{l} \alpha_i K(x, x_i) + b \tag{4}$$

Functions that satisfy Mercer's theorem can be used as kernel functions. We have opted for the RBF kernel function:

$$K(\mathrm{x}, \mathrm{x}_i) = \phi^T(x)\phi(x_i) = \exp(-\|x - x_i\|_2^2 / \sigma^2) \tag{5}$$

The values of c and σ must be pre-determined, when we use the LS-SVM with the RBF kernel function if we wish to make predictions.

3 LS-SVM with Bayesian Evidence Framework

3.1 Level 1 Inference

Given the data points $D = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{l}$, model H (an LS-SVM model with the RBF kernel function), and a given value $\lambda(\lambda = 1/c)$, assuming that the input data are identically distributed independently, we can obtain the model parameters w by maximizing a posteriori probability:

$$p(w \mid D, \lambda, H) = \frac{p(D \mid w, \lambda, H)\, p(w \mid \lambda, H)}{p(D \mid \lambda, H)} \tag{6}$$

We assume the sample data points are independent of each other. It follows that:

$$p(D \mid w, \lambda, H) = \prod_{i=1}^{l} p(x_i, y_i \mid w, \lambda, H) \tag{7}$$

Since $p(x_i, y_i \mid w, \lambda, H) \propto p(\xi_i \mid w, \lambda, H)$ and from (1) we know the error $\xi_i = y_i - (w^T \phi(x_i) + b)$. If we assume the error has a Gaussian distribution, then:

$$p(\xi_i \mid w, \lambda, H) = \sqrt{\frac{\lambda}{2\pi}} \exp(-\frac{1}{2}\xi_i^2) \tag{8}$$

The evidence $p(D \mid \lambda, H)$ is a constant, which will be used at the second inference. Assume that w is a Gaussian distribution, then:

$$p(w \mid \lambda, H) = (\frac{\lambda}{2\pi})^{\frac{l}{2}} \exp(-\frac{\lambda}{2} w^T w) \tag{9}$$

Then, we can substitute (7)-(9) into (6) to obtain the following:

$$p(w \mid D, \lambda, H) \propto \exp(-\frac{\lambda}{2} w^T w - \sum_{i=1}^{l} \xi_i^2) = \exp(-J(w, \xi)) \tag{10}$$

Equation (10) shows that the maximum posteriori estimates w_{MP} is obtained by minimizing the negative logarithm of (1). This corresponds to solving the set of linear (3) in the dual space.

3.2 Level 2 Inference

In the second level of inference, Bayes' rule is applied to infer the maximum a posteriori λ_{MP} values from the given data D. Assume that $p(\lambda \mid H)$ is of flat prior distribution，then we have

$$\begin{aligned} & p(\lambda \mid D, H) = \frac{p(D \mid \lambda, H) p(\lambda \mid H)}{p(D \mid H)} \propto p(D \mid \lambda, H) \propto \\ & \int p(D \mid w, \lambda, H) p(w \mid \lambda, H) dw \\ & = (\frac{\lambda}{2\pi})^{\frac{l}{2}} \int \exp(-\frac{\lambda}{2} w^T w - \frac{1}{2} \sum_{i=1}^{l} \xi_i^2) dw \end{aligned} \tag{11}$$

defining

$$\begin{cases} E_W = \dfrac{1}{2} w^T w \\ E_D = \displaystyle\sum_{i=1}^{l} \frac{1}{2} \xi_i^2 \\ A = \dfrac{\partial^2 (\lambda E_W + E_D)}{\partial w^2} \end{cases} \tag{12}$$

E_W^{MP}, E_D^{MP} is the value of E_W, E_D respectively when $w = w_{MP}$. We take logarithm on both sides of (11), then

$$\ln p(\lambda \mid D, H) \propto -\lambda E_W^{MP} - E_D^{MP} + \frac{l}{2}\ln \lambda - \frac{1}{2}\ln(\det A) + C \tag{13}$$

where C is a constant, the maximization of the log-posterior probability of $p(\lambda \mid D, H)$ with respect to λ leads to the most probable value λ_{MP}, which obtained by the following equation:

$$2\lambda_{MP} E_w^{MP} = \gamma \tag{14}$$

where $\gamma = l - \lambda \text{trace} A^{-1}$, for the least square vector machine

$$\begin{cases} A = \dfrac{\partial^2(\lambda E_W + E_D)}{\partial w^2} = \lambda I + B \\ B = \sum_{i=1}^{l} \phi^T(x_i)\phi(x_i) \end{cases} \tag{15}$$

where I is the identity matrix, $\phi(x_i) = (\phi(x_1), \phi(x_2), \cdots, \phi(x_l))$, $N(N \le l)$ denotes the number of nonzero eigenvalues ρ_i of B, which is $l \times l$ matrix. Then:

$$\gamma = \sum_{i=1}^{N} \frac{\rho_i}{\lambda + \rho_i} \tag{16}$$

We can obtain the optimal regularization parameter λ_{MP} by iterating (14) and (16).

3.3 Level 3 Inference

In level 3 inference of the evidence framework, the posterior probabilities of different models can be examined to find the optimal kernel parameter. Assume that the prior probability over all possible models is uniform, then (17) can be obtained.

$$\begin{aligned} p(H \mid D) &= \frac{p(D \mid H)p(H)}{p(D)} \propto p(D \mid H) \propto \\ &= \int p(D \mid \lambda, H)p(\lambda \mid H)d\lambda \propto \frac{p(D \mid \lambda_{MP}, H)}{\sqrt{\gamma}} \end{aligned} \tag{17}$$

Therefore

$$\ln p(H \mid D) = -\lambda_{MP} E_W^{MP} - E_D^{MP} + \frac{l}{2}\ln \lambda_{MP} + \frac{1}{2}\ln(\det A) - \frac{1}{2}\ln \gamma + C \tag{18}$$

The optimal kernel parameter can be achieved by maximizing log-posterior probabilities $\ln p(H \mid D)$, which is searched with respect to the different kernel parameter in the appropriate ranges.

4 Data and Method

4.1 Data Set

The training set was obtained from ASEdb [10] and the dataset of Cho [6], which was derived from 17 protein–protein complexes. Proteins are considered as

nonhomologous when the sequence identity is no more than 35% and the SSAP [11] score is 80. The sequence identity and SSAP score can be obtained using the CATH query system [12]. If homologous pairs are included, the sites of recognition differ from the two proteins. The atomic coordinates of the protein chains are obtained from the Protein Data Bank (PDB) [13]. The residues with $\Delta\Delta G \geq 2.0$ kcal/mol are defined as hot spots, and those $\Delta\Delta G < 0.4$ kcal/mol with are regarded as non-hot spots. The other residues are not included in the training set in order to gain better discrimination. The final two-class training set contains 158 interface residues, of which 65 are hot spots and 93 are non-hot spots.

In addition, an independent test set is constructed from the BID [14] to further validate our proposed model. In the BID database, the alanine mutation data are listed as 'strong', 'intermediate', 'weak' and 'insignificant'. In our study, only 'strong' mutations are considered as hot spots; the other mutations are regarded as energetically unimportant residues. This test set consists of 18 complexes, where 127 alanine-mutated data are contained and 39 residues are hot spots.

4.2 Features Extraction

Based on the previous studies about hot spots prediction, we generate 10 physicochemical features and 49 structure features.

The physicochemical features used in the experiment include the number of atoms, the number of electrostatic charge, the number of potential hydrogen bonds, hydrophobicity, hydrophobicity, propensity, isoelectric point, mass, the expected number of contacts within 14Å sphere, and electron-ion interaction potential. These features were only related to the amino acid types and no structural information is contained.

The structure features include accessible surface area (ASA), relative ASA, depth index (DI), and protrusion index (PI). From ASA and RASA, five derived attributes are total (total sum of all atom values), backbone (the sum of all backbone atom values), side-chain (the sum of all side-chain atom values), polar (the sum of all oxygen, nitrogen atom values) and non-polar (the sum of all carbon atom values). Based on DI and PI, four residue attributes can be obtained as total mean (the mean value of all atom values), side-chain mean (the mean value of all side-chain atom values), maximum (the maximum of all atom values) and minimum (the minimum of all atom values). Therefore, the structure information was generated by PSAIA from both the unbound and the bound state.

In addition, the relative changes of ASA, DI and PI between the unbound and the bound states of the residues were calculated as Xia [7].

4.3 Feature Selection

Feature selection is a key step ahead of classifiers designing, by which we can avoid overfitting, improve model performance and provide faster and more effective models. In present study, feature selection is performed with a two-step method, which integrates both the filter and the wrapper, and shows the best subset of features for discriminating hot spots from other residues.

Initially, we assess the feature vector elements using the F-score [15], which assesses the discriminatory power of each individual feature. The F-score is calculated as follow.

$$F(i) = \frac{(\bar{x}_{ni} - \bar{x})^2 + (\bar{x}_{hi} - \bar{x})^2}{\sigma_{ni} + \sigma_{hi}} \tag{19}$$

where $\bar{x}_{ni}$, $\bar{x}_{hi}$ and $\bar{x}$ are the mean of the non-hot spots , the mean hot spots and the mean in the whole data set, respectively. σ_{ni} and σ_{hi} are the corresponding standard deviations.

Secondly, we use a wrapper-based feature selection strategy where features are evaluated by 10-fold cross-validation performance, and redundant features are removed by sequential backward elimination (SBE). The SBE scheme sequentially removes features from the whole feature set until the optimal feature subset is obtained. Each removed feature is one that maximizes the performance of the predictor. The ranking criterion represents the prediction performance of the predictor, which is built by a subset features exclusive of feature and is defined as follow:

$$R(i) = \frac{1}{k}\sum_{j=1}^{k}(AUC_j + A_j + R_j + P_j) \tag{20}$$

where k is the repeat times of 10-fold cross validation; AUC_j, A_j, R_j and P_j represent the values of AUC score, accuracy, recall and precision of the 10-fold cross validation, respectively.

As a result, a set of 9 optimal features are obtained. We find that the structural properties dominate the top list. Thus it can be concluded that structural properties are more predictive than physicochemical properties in determining hot spot residues. The proposed framework is sketched in Fig. 1.

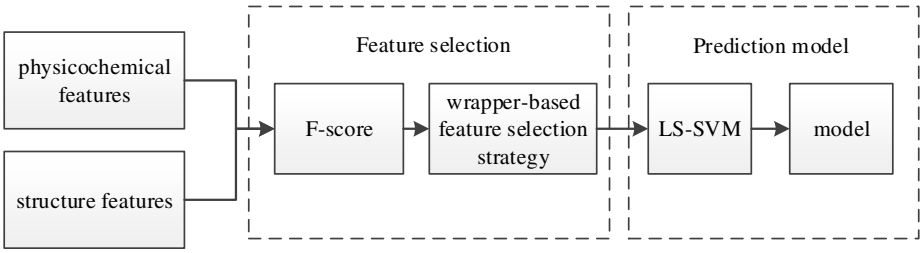


Fig. 1. The framework of hot spots prediction. Firstly, we generate 10 physicochemical features and 49 structure features. Then we apply a two-step method to select effective features. Finally, we employ LS-SVM in Bayesian inference to construct the prediction model.

4.4 Performance Evaluation

In our experiments several measures are used to evaluate the performance of these classifiers including Precision (P), Recall (R), F-measure (F1). These measures are defined as follows:

$$P=\frac{TP}{TP+FP} \tag{21}$$

$$R=\frac{TP}{TP+FN} \tag{22}$$

$$F1=\frac{2PR}{P+R} \tag{23}$$

where TP, FP, TN, FN refer to the number of true positive, false positive, true negative, false negative, respectively.

Recall is the proportion of number of correctly classified hot spot residues to the number of all hot spot residues. Precision is the ratio of number of correctly classified hot spot residues to the number of all residues classified as hot spots. F-measure (F1) is a measure to balances precision and recall.

For practical significance, F1-score has to exceed the frequency of hot spots observed in the data set. As the training set consists of 65 hot spots and 93 non-hot spot residues, the F1-score for any model should be more than 0.58. For the independent test set, the F1-score should be larger than 0.47.

5 Experimental Results

We test the LS-SVM-Bayesian model by the 10-fold cross validation in the dataset. The dataset is randomly partitioned into 10 mutually exclusive subsets of nearly equal size. The 9-fold is used as a training set and the remaining 1-fold is used as a test set. To evaluate the performance of our method, the existing hot spots prediction method Robetta [16], FOLDEF [17], MINERVA2 [6], HotPoint [4], KFC [5] are implemented and evaluated on Dataset set with 10-fold cross-validation. We use the estimated $\Delta\Delta G$ value as the classification score. The performance of each model is measured by three metrics.

Table 1. Performance comparison on the training set

Method	Recall	Precision	F1	ΔF1
Robetta	0.51	0.89	0.65	**
FOLDEF	0.31	0.91	0.46	-0.19
MINERVA2	0.58	**0.93**	0.72	0.07
HotPoint	0.54	0.73	0.62	-0.03
KFC	0.55	0.75	0.64	-0.01
Our method	**0.88**	0.81	**0.84**	**0.19**

Table 1 is the detailed results by comparing our method with the existing methods in the training set. Although, our method has the lower precision than Robetta, FOLDEF and MINERVA2, our method performs best in two performance metrics (Recall=0.88 and F1-score=0.84), which shows that our method can predict correctly more hot spots and has better balance in prediction performance than the existing methods.

Table 2. Performance comparison on the test dataset

Method	Recall	Precision	F1	ΔF1
Robetta	0.33	0.52	0.41	**
FOLDEF	0.26	0.48	0.33	-0.08
MINERVA2	0.44	**0.65**	0.52	0.11
HotPoint	0.59	0.50	0.54	0.13
KFC	0.31	0.48	0.38	-0.03
Our method	**0.67**	0.51	**0.58**	**0.17**

In addition, we further validate the performance of the proposed model on the independent test dataset. Results of the independent test are presented in Table 2. From this table, we know that our method show the highest Recall (Recall=0.67), which outperforms all other method. Especially, our method's recall is 14% higher than that of HotPoint, which has the highest sensitivity among the existing methods. This means that our method can predict more hot spots and is helpful for the identification of hot spots residues in the practical applications. Also, the F1 score of our method is 7% higher than that of HotPoint. From the analyses above, we find that our method can obtain better prediction performance in comparison to other available prediction approaches.

6 Conclusion

In this study, we have described an efficient method, LS-SVM in Bayesian inference to predict hot spot residues in protein interfaces with low computation cost. Compared with the previous models, our method appears to be significant performance in recall as well as F1 score that measures the balance between precision and recall. As for the future work, we will explore more useful features in both hot spots and non-hot spots. What's more, we will try to improve the proposed method with the recent advanced machine learning techniques.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China (61273225, 61273303), the Open Foundation (2010D11) of State Key Laboratory of Bioelectronics, Southeast University, the Program of Wuhan Subject Chief Scientist (201150530152), as well as National "Twelfth Five-Year" Plan for Science & Technology Support (2012BAC22B01).

References

1. Albert, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: Essential Cell Biology, 3rd edn. Galand Science (2010)
2. Cosic, I.: The resonant recognition model of macromolecular bioactivity: theory and applications. Bikrhauser Verlag (1997)

3. Bogan, A.A., Thorn, K.S.: Anatomy of hot spots in protein interfaces. Journal of Molecular Biology 280(1), 1–9 (1998)
4. Tuncbag, N., Kenskin, O., Gursoy, A.: HotPoint: hot spot prediction server for protein interfaces. Nucleic Acids Research 3(suppl. 2), W402–W406 (2010)
5. Darnell, S., LeGault, L., Mitchell, J.: KFC Server: interactive forecasting of protein interaction hot spots. Nucleic Acids Research, W265–W269 (2008)
6. Cho, K.-I., Kim, D., Lee, D.: A feature-based approach to modeling protein-protein interaction hot spots. Nucleic Acids Research 37(8), 2672–2687 (2009)
7. Xia, J.-F., Zhao, X.-M., Song, J., Huang, D.-S.: APIS: accurate prediction of hot spots in protein interfaces by combining protrusion index with solvent accessibility. BMC Bioinformatics 11(174), 1471–2105 (2010)
8. Zhang, S., Zhang, X.: Prediction of Hot Spot at Protein-Protein Interface. Acta Biophysica Sinica 29(2), 151–157 (2013)
9. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
10. Thorn, K.S., Bogan, A.A.: ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions. Bioinformatics 17(3), 284–285 (2001)
11. Toth, G., Watts, C.R., Murphy, R.F., Lovas, S.: Significance of aromatic-backbone amide interactions in protein structure. Protein Struct. Funct. Genet. 43, 373–381 (2001)
12. Pearl, F.M., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., Bennett, C., Marsden, R., Grant, A., Lee, D., et al.: The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucleic Acids Res. 33(suppl. 1), D247–D251 (2005)
13. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Res. 28(1), 235–242 (2000)
14. Fischer, T., Arunachalam, K., Bailey, D., Mangual, V., Bakhru, S., Russo, R., Huang, D., Paczkowski, M., Lalchandani, V., Ramachandra, C.: The binding interface database (BID): a compilation of amino acid hot spots in protein interfaces. Bioinformatics 19(11), 1453–1454 (2003)
15. Chen, Y.W., Lin, C.J.: Combining SVMs with various feature selection strategies [EB/OL] (August 10, 2009), http://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf
16. Kortemme, T., Baker, D.: A simple physical model for binding energy hot spots in protein-protein complexes. Proceedings of the National Academy of Sciences of the United States of America 99(22), 14116–14121 (2002)
17. Guerois, R., Nielsen, J., Serrano, L.: Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. Journal of Molecular Biology 320(2), 369–387 (2002)

Predicting the Survival Status of Cancer Patients with Traditional Chinese Medicine Symptom Variation Using Logistic Regression Model

Min Wan[1], Liying Fang[1], Mingwei Yu[2], Wenshuai Cheng[1], and Pu Wang[1]

[1] College of Electronic Information and Control Engineering,
Beijing University of Technology, Beijing 100124, China
[2] Beijing Hospital of Traditional Chinese Medicine, Beijing 100010, China
wanmin625@139.com, fangliying@bjut.edu.cn,
yumingwei1120@163.com, 409325865@qq.com, wangpu@bjut.edu.cn

Abstract. Background- In the study of rules in pathological changes, most of the traditional analyses are from the static perspective, which regard cross-sectional data as the input dataset to analyze the effect of non-time-varying factors. However, according to the clinical experiences, the changes of physical status can partly reflect the disease progression and the mortality which have been ignored in the existing studies. Thus, based on the dynamic perspective, by using the changing patterns of symptom as the model input, longitudinal data can be utilized to further explore the rules in pathological changes from the dynamic perspective which is a novel and effective solution. **Method**- The study proposed a dynamic pattern representation method; pretreated and transformed the original dataset, including the Traditional Chinese Medicine (TCM) and the western medicine clinical longitudinal data, into a 2-dimensional matrix composed of the symptom indexes and the changing patterns; and analyzed the influences between the changing patterns of symptom and III stage non-small cell lung cancer (NSCLC) patient's mortality by multivariate logistic regression. **Result**- The predicting accuracy using the transformed dataset by proposed representation method is 90.7%. Based on the enter stepwise regression method, the accuracy increased 26.3% and 14.5% than the baseline dataset and the last records respectively; based on the forward stepwise regression method, the accuracy increased 16.7% and 3% than the baseline dataset and the last records respectively. **Conclusion**- The experiment results indicated that the proposed data representation method is feasible and effective, meanwhile, the proposed novel dynamic perspective appears more appropriate for the TCM mainly III stage NSCLC patients' modeling than the traditional static method.

Keywords: Longitudinal data, Logistic regression, Traditional Chinese Medicine symptom, Cancer.

1 Introduction

Longitudinal data is prevalent in biological and social science, in which measurements have been made repeatedly on a cohort of subjects at a sequence of time points or other

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 211–218, 2013.

condition. Unifying the merits of cross-sectional and time-series data, longitudinal data can imply both the interaction between the independent variables and the dependent variables, and indicate co-relationships and embedded dynamic change information on the time axis [1]. During clinical cancer treatment, a patient's follow-up records form a group of longitudinal data, which includes massive symptoms, interventions and period evaluations. Nowadays, the therapies of western medicine in cancer mainly include surgery, radiotherapy and chemotherapy, which are important methods in locally control of tumor growth. But, the postoperative progression is not optimistic; high recurrence and transport rate puzzle patients and clinicians. On the other hand, TCM emphasize on the concept of the wholism, not only limited to the lesion itself, but also pay much attention to the patients' physiological function by macro regulation. In China, the history of the combination between TCM and western medicine of tumor is over 40 years, which can effectively relieve patients' suffering and improve the life quality. Also, it has been widely accepted by patients, and received the recognition from the cancer community. A large number of clinical practices have proved that the curative effects of the combination were significantly better than unitary western medicine or TCM treatment, especially in improving the life quality of terminal-stage cancer patients. Through preliminary research, statistical analysis model and data mining technology have been well applied for longitudinal data analysis in the field of the western medicine [2-7]; however, modeling for TCM data is still in an early stage [8-11]. Furthermore, combined with TCM clinical experience, a clinician can judge tumor progress or death possibility by comparing patients' changing patterns of physical statuses subjectively. In another word, without statistic basis, the TCM clinician can only conclude the correlation between the change of longitudinal clinical data and cancer progress by their own experience roughly. Due to the characteristics of the longitudinal data, there are two kinds of changes can be explored: the interaction variability of cross-sectional data and the correlation variability of time-series data [12, 13]. TCM data is normally descriptive and categorical type, while western medicine data is numeric type; however, most models can only accept a single type of data as the input; thus, how to analyze TCM and western medicine data simultaneously is one of the challenges that should be further considered. Moreover, the description of the TCM characteristics is usually based on clinicians' own experiences and habits. Therefore, how to achieve the unifying and coding of TCM data is another challenge. In addition, among various kinds of clinical TCM indicators from four diagnostic methods, which kind of factors can directly reflect the change of the disease is unknown. As a consequence, tumor progress trend can be explored by analyzing the clinical longitudinal data; the changing patterns of physical statuses can be expressed by time-series data feature, in another aspect, the effect weight of different symptoms can be indicated by cross-sectional data feature.

Generalized linear model is a kind of promotion of general linear model by breaking through the limitation that the variables should belong to the normal distribution, and expands the variable distribution to the exponential distribution family (Binomial distribution, Poisson distribution and Negative binomial

distribution, etc.) [1, 14]. By specifying different connecting functions, the transformed dependent variables can satisfy the requirements of linear model, and the linear model analysis methods are competent to solve model structure, parameter estimation and model evaluation problems; meanwhile, the exponential distribution family model are unified into the generalized linear models framework. Within the scope of the generalized linear model, Logistic regression model treats disease as dependent variables and rules in pathological changes as independent variables, and estimates relative risk or odd ratio of each factor, which is not only suitable for cohort data study but also case-control study.

In this paper, a dynamic pattern representation method was proposed, and a TCM-based Logistic regression model was established to analyze the relationship between the changing patterns of symptom and the III stage non-small cell lung cancer (NSCLC) patient's mortality. Different from traditional risk factors analysis model, the proposed method emphasizes on the dynamic perspective, by considering the independent variable changing characteristics via time axis, and gives more convincing statistic result for medical assist support.

2 Experiment and Method

2.1 Variable Pretreatment

There are 2233 original clinical follow-up records from a cohort of 216 III stage NSCLC patients, recorded from January 2008 to January2010. In the selected data set, after abandoned the expulsion records, the average case of sampling point is 4 to 14, and the unequal length of sampling points was due to the clinical characteristics, such as different starting time, death, etc. For those death patients, the next-to-last record was used and regarded as the last. In the multivariate analysis, the input dataset including all-recorded 17 TCM symptoms: cough, expectoration, blood-stained sputum, breathe hard, choking sensation in chest, chest pain, dry throat, fever, mentally fatigued, Anorexia, spontaneous perspiration and night sweat, insomnia, constipation, diarrhea, more frequent urination, dysphoria in chestpalms-soles as well as extreme chilliness; 3 common western medicine symptoms: tumor specific growth factor, carcino-embryonic antigen and carbohydrate antigen 125.

By analyzing the relationship between the changing patterns of physical statuses and the survival condition, the target of the method is demonstrated that the dynamic perspective performs better than the static perspective in explaining the modeling rules in pathological changes using the combined TCM and western medicine data. Thus, in order to realize the dynamic procedure expression, the variable changing method was proposed to simplify the input variables type and unify the expression way. In the original dataset, 4 levels were used to describe the TCM symptoms severity, "0" means the patient does not have that symptom; "1" to "3" means the severity of that symptom sequentially increasing. While, western clinical data are mainly numeric data that need to be normalized. In order to unify the requirements of the model inputs, coding rules were established and the input data were all transformed into 4 levels categorical type that ranging from "0" to "3". By comparing

the dynamic changes between patients' baseline (first time check during the follow-up period) and last (last time check during the follow-up period) records, a new dataset with changing classification edition was transformed from a sequence of follow-up records, and the coding rule of transforming process is shown in Table 1. In the transformed dataset, the distance between the end status and the begin status of a follow-up records sequence is applied in this paper, in order to depict the changing degree of the sequence. In Table 1, "0" to "3" means the symptom did not change from the baseline records to the last time diagnose; "4" and "6" means the symptom changed 1 level (increase or decrease); "5" and "7" means the symptom changed more than 2 levels (increase or decrease). Thus, there are 8 possible values for each variable.

Table 1. Variable transformation rule

Begin-End status	0-0	1-1	2-2	3-3	Rank adding =1	Rank adding >=2	Rank decreasing =1	Rank decreasing >=2
Transformed variable description	0	1	2	3	4	5	6	7

2.2 Experiment Design

In order to demonstrate that the dynamic change pattern is more adaptable than the cross sectional data in analyzing the clinical progression, predicting the mortality, and selecting variables which can reflect the clinical result directly, the contrast experiments were designed. Firstly, based on the Logistic model, using the same stepwise method, with the patients' survival condition as output, different datasets (the baseline data only, the last records only and the transformed data) were utilized to compare the most appropriate input type. Secondly, based on the same prerequisite, different stepwise methods (enter and forward) were used to choose the highest accuracy input dataset.

2.3 Statistical Analysis

In the study, multivariate analyses were performed by logistic regression method, which is suitable for classification type variables. The input dataset was transformed from the original combined dataset into a two-dimensional matrix with 216 lines and 20 columns; the columns represent 20 symptoms, the lines represent the changing patterns of physical statuses, and each variable has 8 possible categorical type values, instead of original descriptive and numeric type. By establishing Logistic regression model, the effect coefficient between 20 independent variables and the dependent variables can be explored. In the statistical process, $P<0.05$ was considered statistically significant.

3 Results

3.1 Multivariate Analysis Using Logistic Regression Modeling

Based on the III stage NSCLC patients' clinical follow-up longitudinal data, the study mainly focused on identifying the changing patterns of physical statuses, and established a Logistic regression model to analyze the relationship between the TCM mainly symptom changing patterns and the mortality. After 10-fold cross validation, the results in Table 2 show that the model accuracy of different input datasets (the baseline data, the last records and the transformed data) and stepwise methods (enter and forward). The proposed transformed data with enter stepwise method performed the best; the accuracy reached 90.7%, and indicated that the combined TCM and western medicine symptoms influenced the mortality significantly. Based on different stepwise methods, the performance of the proposed transformed data was better than other ordinary datasets; there are 26.3% and 14.5% accuracy increasing compared with the baseline data and the last records by enter stepwise method; meanwhile, there are 16.7% and 3% increasing than ordinary datasets respectively by forward stepwise method. By combining three kinds of input dataset and two stepwise approaches, the accuracies are the transformed data with enter, the last record with enter, the transformed data with forward, the baseline data with enter, the last record with forward, the baseline data with forward in descending order.

Table 2. Modeling results of Logistic regression based on TCM symptom changing patterns of physical statuses

Regression mode	Change_ Enter	Change_ Forward	Baseline_ Enter	Baseline_ Forward	Last_ Enter	Last_ Forward
Classification accuracy	90.7%	74%	71.8%	63.4%	79.2%	71.8%

Figure1 shows the comparison of the specificity and the sensitivity of different datasets and stepwise methods. From the bar chart, we can conclude that the specificity of the proposed transformed dataset with enter stepwise method is the highest, which achieves 85.7. Meanwhile, the sensitivity of the proposed transformed dataset with enter stepwise method is 93.9, which is a little higher than the result of the baseline data under forward stepwise method, that is 93.2. Besides, the overall results also indicate that the proposed transformed dataset can better indicate the effect of the pathogenic factor on the survival condition.

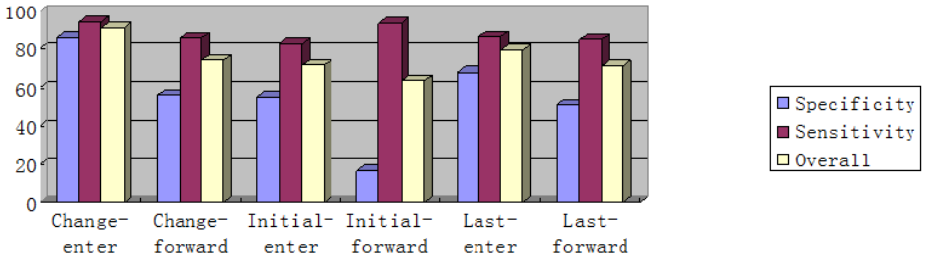
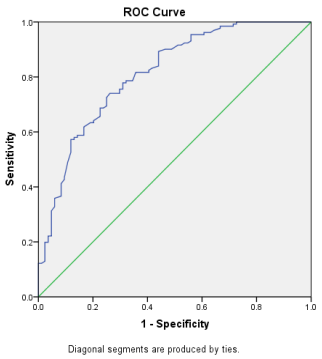
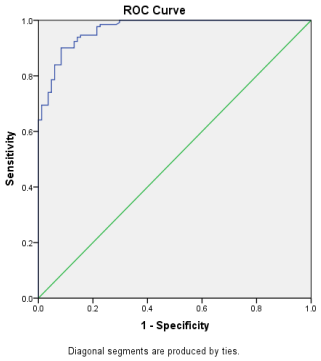
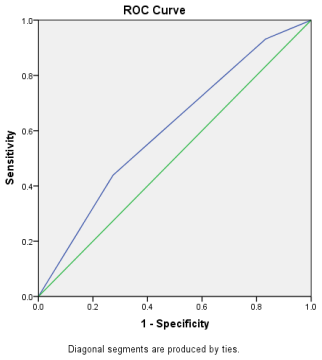
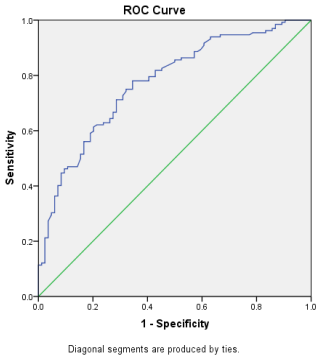
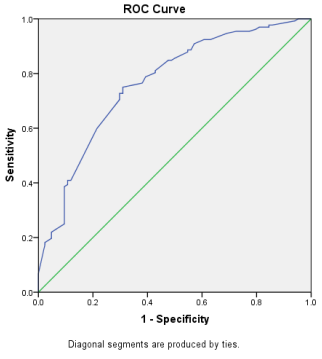
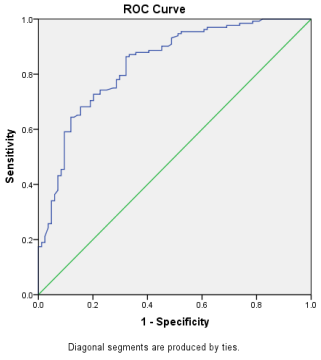


Fig. 1. Comparing results of specificity and sensitivity among different methods

Table 3. ROC curve of different methods

	
Forward: Conditional (Transformed Dataset)	Enter (Transformed Dataset)
	
Forward: Conditional (Baseline dataset)	Enter (Baseline dataset)
	
Forward: Conditional (Last dataset)	Enter (Last dataset)

Receiver-Operating Characteristic (ROC) curve is used for model verification, and the results are shown in Table 3. When utilizing the proposed changing dataset as input with enter stepwise method, the area under the ROC curve is 0.968 (std.Error 0.010), while utilizing the proposed changing dataset as input with forward stepwise method, the area under the ROC curve is 0.813(std.Error 0.030). According to the

statistic results, we can conclude that the enter stepwise method leaded to a higher classification performance for the specific subject. However, if using cross-sectional data as input and ignoring the co-relationship variability of time-series data, when using the baseline dataset with enter stepwise method, the area under the ROC curve is 0.774(std.Error 0.032); when using the baseline dataset with forward stepwise method, the area under the ROC curve is 0.605(std.Error 0.039) . The statistic results indicated that the classification performance of utilizing the proposed changing dataset is higher than using the baseline dataset. Moreover, the area under the ROC curve is 0.838(std.Error 0.028) and 0.765(std.Error 0.033) when using the last records dataset with enter and forward stepwise method respectively. Although the classification results performed better than using the baseline dataset, which were still lower than utilizing the proposed changing dataset. In summary, the results of ROC curves show that the proposed dynamic pattern representation approach can effectively achieve a better performance of modeling accuracy for medical support.

4 Discussion

Currently, the combination between TCM and western medicine of tumor has been gradually accepted by patients and clinicians. The various kinds of clinical TCM indicators from four diagnostic methods and the tumor progression are related to each other. Focused on III stage NSCLC patients, this paper mainly used the clinical TCM changing patterns of physical statuses as input data to evaluate the mortality; and the model accuracy was over 90%. The results not only indicated that the influence of TCM data to lung cancer should be further explored, but also proved that the patient with the same period of the disease (stage III NSCLC) may lead to different living statuses if different changing patterns of physical statuses appears. Comparing with the baseline data or the last follow-up records as model input, focusing on the dynamic change can better predict the classification results. Moreover, unifying the merits of cross-sectional and time-series characteristics in longitudinal data, the derivation of the dynamic factor into medical modeling, focusing on the changing patterns of various kinds of physical statuses with many patients, the research on the rules in pathological changes with transformed input dataset is regarded as a novel idea. Furthermore, the experiment results showed that the proposed pretreatment method, representing dynamic process by comparing the baseline and the last record, coding the symptoms according to the grade change span and describing in eight categories, is feasible and effective. The proposed model can be applied to clinical diagnosis and treatment, and the results have to be considered as exploratory results rather than definitive clinical predictions. According to the patients' existing symptoms, clinician assumes several possible disease progressions and uses the model to predict the mortality. The results can assist the clinician not only to explain disease progress trends and possible results, but also to adjust and improve treatment strategies to a lower mortality condition.

At present, only the baseline data and the last follow-up statues have been taken into comparison as a basis for the proposed changing dataset, but it is not sufficient to

indicate complete embedded information contained in the entire time sequence. In the next step, all follow-up sequence should be taken into consideration by cluster analysis. Based on the cluster principle, various kinds of time series changing form can be unified into several clusters; thus the correlation between the clustered patterns and the prediction target can be discovered.

References

1. Liang, K.Y., Zeger, S.T.: Longitudinal data analysis using generalized linear models. Biometrics 73(1), 13 (1986)
2. Etta, D.P., Suddhasatta, A., Elodia, B.C., et al.: Cancer Cases from ACRIN Digital Mammographic Imaging Screening Trial: Radiologist Analysis with Use of a Logistic Regression Model1. Radiology 2(252), 348–357 (2009)
3. Stephenson, A.J., Smith, A., Kattan, M.W., et al.: Integration of Gene Expression Profiling and Clinical Variables to Predict Prostate Carcinoma Recurrence after Radical Prostatectomy. Cancer 2(104), 290–298 (2005)
4. Lixu, Q., Simon, X.Y., Frank, P., et al.: Modelling and risk factor analysis of Salmonella Typhimurium DT104 and non-DT104 infections. Expert Systems with Applications 35, 956–966 (2008)
5. Li, G., Zhu, Y., Zheng, W., et al.: Analysis of factors influencing skip lymphatic metastasis in pN2 non-small cell lung cancer. Chinese Journal of Cancer Research 4(24), 340–345 (2012)
6. Laurence, S.F., Bernice, O., Siegal, S.: Using Time-dependent Covariate Analysis to Elucidate the Relation of Smoking History to Warthin's Tumor Risk. American Journal of Epidemiology 170(9), 1178–1185 (2009)
7. Shigeyuki, M., Richard, S., Pingping, Q., et al.: Developing and Validating Continuous Genomic Signatures in Randomized Clinical Trials for Predictive Medicine. Predictive Biomarkers and Personalized Medicine 21(18), 6065–6073 (2012)
8. Wang, B., Zhang, M., Zhang, B., et al.: Data mining application to syndrome differentiation in traditional Chinese medicine. In: 7th International Conference on Parallel and Distributed Computing, Application and Technologies, pp. 128–131. IEEE Press, Taipei (2006)
9. Yang, F., Tang, G., Jin, H.: Knowledge mining of Traditional Chinese Medicine Constitution classification rules based on Artificial Fish School Algorithm. In: 3th International Conference on Communication Software and Networks (ICCSN), pp. 462–466. IEEE Press, Xi'an (2011)
10. Pan, X., Zhou, X., Song, H., et al.: Enhanced data extraction, transforming and loading processing for traditional Chinese medicine clinical data warehouse. In: 14th International Conference on e-Health Networking, Applications and Services, pp. 57–61. IEEE Press, Beijing (2012)
11. Wang, Y., Ma, L., Liao, X., et al.: Decision tree method to extract syndrome differentiation rules of posthepatitic cirrhosis in traditional Chinese medicine. In: IEEE International Symposium on IT in Medicine and Education, pp. 744–748. IEEE Press, Xiamen (2008)
12. Lawless, J.F.: Statistical Models and Methods for Lifetime Data. Wiley, New York (1982)
13. Breslow, N.E., Day, N.E.: Statistical Methods in Cancer Research. IARC, Lyon (1987)
14. Cox, D.R., Oakes, D.: Analysis of Survival Data. Chapman & Hall, London (1984)

Exploiting Multiple Features for Learning to Rank in Expert Finding

Hai-Tao Zheng[1], Qi Li[1], Yong Jiang[1], Shu-Tao Xia[1], and Lanshan Zhang[2]

[1] Tsinghua-Southampton Web Science Laboratory,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[2] Digital Media Art and Design Institute,
Beijing University of Posts and Telecommunications
{zheng.haitao,jiangy,xiast}@sz.tsinghua.edu.cn,
purgiant@gmail.com, zls326@sina.com

Abstract. Expert finding is the process of identifying experts given a particular topic. In this paper, we propose a method called Learning to Rank for Expert Finding (LREF) attempting to leverage learning to rank to improve the estimation for expert finding. Learning to rank is an established means of predicting ranking and has recently demonstrated high promise in information retrieval. LREF first defines representations for both topics and experts, and then collects the existing popular language models and basic document features to form feature vectors for learning purpose from the representations. Finally, LRER adopts RankSVM, a pair wise learning to rank algorithm, to generate the lists of experts for topics. Extensive experiments in comparison with the language models (profile based model and document based model), which are state-of-the-art expert finding methods, show that LREF enhances expert finding accuracy.

Keywords: Expert finding, Language model, Learning to rank, Features.

1 Introduction

In enterprise or common web search settings users not only search valuable documents but also interested experts. Most important documents are composed by these people, and they can answer some detailed questions. For example, a department of a company may look for experts from the company to assist them in formulating a plan to solve a problem [3–8]. Organizers of a conference behoove assign submissions to the Program Committee (PC) members based on their research interest and expertise [9]. In some cases, to find an expert is critical. A patient in danger needs to find a professional doctor to diagnose his desease and to treat it. Currently, people have to manually identify the experts, which is obviously laborious, time-consuming and expensive. It is of significant value to study how to identify experts for a specific expertise area automatically and precisely.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 219–230, 2013.

An expert finding system is an information retrieval (IR) system that can aid users with their "expertise need" in the above scenarios. The system helps to find individuals or even working groups possessing certain expertise and knowledge within an organization or an association network. As a retrieval task, expert finding was launched as a part of the annual enterprise track [9–11, 13–16] at the Text REtrieval Conference (TREC) [17] in 2005 (TREC,2005) and has recently attracted much attention. An active research problem is how best to generate a ranking of candidates from a collection of documents and many heuristic models have been proposed to tackle the problem [13, 11, 16, 8, 1, 2]. We try to intermix these models to improve accuracy. However, most existing modeling approaches have not sufficiently taken into account document features. Besides, expert finding can be considered as a specific problem of information retrieval and there have been many carefully devised models for the traditional ad-hoc information retrieval. We assume that expert finding can benefit from the incorporation of document features. Therefor, these features are expected to blend together to give a better solution. To implement what we envision, we make use of learning to rank [21, 19] to combine all these models and document features effectively. Learning to rank is a machine learning framework that can subsume most proposed models to generate a better ranking and has also become a very hot research direction in information retrival in recent years.

In this paper we introduce learning to rank for expert finding (LREF) to enhance the ranking performance of expert finding. To provide insights into the relation between topics and experts, we propose representations for topics and experts, which can be a foundation for most existing expert finding methods. In this way, new features based on new proposed models can be easily extended to LREF. Then profile- and document-based features including TF, TF-IDF, language models and so on are extracted from the representations. After that, RankSVM, a learning to rank algorithm, is chosen to testify the effectiveness and experiments show that LREF incorporating multi-features performs better than the candidate language model and the document language model. To the best of our knowledge, LREF is the first attempt that adapts learning to rank to expert finding.

The structure of this paper is organized as follows. We discuss related work of expert finding in section 2. Section 3 is devoted to a detailed description of our method to address the task. We connect documents to topics and experts separately, and use topics and relevant documents as logical queries to search on logical documents which comprise experts and relevant documents. Profile-based features and document-based features are extracted from logical queries and logical experts. At the end of the section we describe the LREF algorithm. Then our experimental setup and experimental evaluations of our model are presented in section 4. Finally, we provide concluding remarks and suggestions for future work in Section 5.

2 Related Work

There are two basic approaches to solve the problem [11, 9, 12]. So far most existing approaches are derivations of these two kinds. The first approach is profile-based. All documents related to a candidate expert are merged into a single personal profile prior to retrieval time. When a query comes, standard retrieval systems measure the relevance between the query and the personal profiles and then return corresponding best candidates to the user. The second approach, document-based, depends on individual documents. When a query comes, the method runs the query against all documents and ranks candidates by summarized scores of associated documents or text windows surrounding the person's mentioning. Document-based methods are claimed to be much more effective than profile-based methods, probably due to the fact that they estimate the relevance of the text content related to a person on the much lower and hence less ambiguous level.

These two approaches can be formulated by generative probabilistic models. Critical to these models is the estimation of the probability of the query topic being generated by the candidate expert. In other words, how likely would this query topic can be talked/written about by the candidate? These two approches to expert finding lead to different language models.

Expert finding addresses the task of finding the right person(s) with the appropriate skills and knowledge:" Who are the experts on topic X?" Put differently: what is the probability of a candidate ca being an expert given the query topic q? Language models will rank the candidates according to the probability.

Formally, suppose $S = \{d_1, \ldots, d_{|S|}\}$ is a collection of supporting documents. Let $q = t_1, t_2, \ldots, t_n$ be the description of a topic, where t_i is a term in the description. Let ca be an expert candidate. The task is how to determine $p(ca|q)$, and rank candidates ca according to this probability. With the higher probability the candidates are the more likely experts for the given topic. Obviously, the challenge is how to accurately estimate the probability $p(ca|q)$. By Bayes' Theorem, we can get

$$p(ca|q) = \frac{p(q|ca) \cdot p(ca)}{p(q)}, \tag{1}$$

where $p(ca)$ is the probability of a candidate and $p(q)$ is the probability of a query. We can ignore $p(q)$ for the ranking purpose, because $p(q)$ is a constant for a given topic. Thus probability of a candidate ca being an expert given the query q is proportial to the product of $p(q|ca)$ and $p(ca)$:

$$p(ca|q) \propto p(q|ca) \cdot p(ca). \tag{2}$$

The language models' main task is to estimate the probability of a query given the candidate, $p(q|ca)$, because this probability represents the extent to which the candidate knows about the query topic. The candidate priors, $p(ca)$, are generally assumed to be uniform, and so they won't influence the ranking. Following are the two primitive language models for expert finding:

Candidate Model: Candidate model (models 1 and 1B in [11], profile-based estimation in [9]) is also called profile-based language model. It builds on well-known intuitions from standard language modeling techniques applied to document retrieval to estimate the probability of a query given a candidate, $p(ca)$. Usually, we use multinomial probability distribution over the vocabulary of terms to represent a candidate expert ca, then infer a candidate model θ_{ca} for each candidate ca. We can get the probability of a term given the candidate model $p(t|\theta_{ca})$ and the probability of a query given the candidate model:

$$p(q|\theta_{ca}) = \sum_{t \in q} p(t|\theta_{ca})^{n(t,q)}, \tag{3}$$

where $n(t, q)$ is the number of times term t appears in query q. Here each term is assumed to be sampled identically and independently. To gain an estimate of $p(ca)$, we can gain an estimate of the probability of a term given a candidate, $p(t|ca)$, which is then smoothed to ensure that there are no non-zero probabilities due to data sparsity:

$$p(t|\theta_{ca}) = (1 - \lambda_{ca}) \cdot p(t|ca) + \lambda_{ca} \cdot p(t), \tag{4}$$

where $p(t)$ is the probability of a term in the document repository. To estimate $p(t|ca)$, we use the documents as a bridge to connect the term t and candidate ca:

$$p(t|ca) = \sum_{d \in D_{ca}} p(t|d, ca) \cdot p(d|ca), \tag{5}$$

Besides, we assume that document and the candidate are conditionally independent, namely $p(t|d, ca) \approx p(t|d)$. Finally, we put equations above together and get:

$$p(q|\theta_{ca}) = \prod_{t \in q} \left\{ (1 - \lambda_{ca}) \cdot \left(\sum_{d \in D_{ca}} \cdot p(d|ca) \right) + \lambda_{ca} \cdot p(t) \right\}^{n(t,q)}. \tag{6}$$

where λ_{ca} is a general smoothing parameter.

Document Model: Instead of creating a profile-based representation of a candidate, documents are modeled (models 2 and 2B in [11], document-based estimation in [9]) and queried, then the candidates associated with the documents are considered possible experts. Like candidate model, we also assume that query terms are sampled identically and independently. By taking the sum over all documents $d \in D_{ca}$, we obtain $p(q|ca)$:

$$p(q|ca) = \sum_{d \in D_{ca}} p(q|d, ca) \cdot p(d|ca) = \sum_{d \in D_{ca}} \prod_{t \in q} p(t|d, ca)^{n(t,q)} \cdot p(d|ca), \tag{7}$$

$(p|ca)$ can be computed by assuming conditional independence between the query and the candidate, that is $p(t|d, ca) \approx p(t|\theta_d)$. For each document d a

document model θ_d is infered, so that the probability of a term t given the document model θ_d is $p(t|\theta_d) = (1 - \lambda_d) \cdot p(t|d) + \lambda_d \cdot p(t)$. Finally we get:

$$p(q|ca) = \sum_{d \in D_{ca}} \prod_{t \in q} \{(1 - \lambda_d) \cdot p(t|d) + \lambda_d \cdot p(t)\}^{n(t,q)} \cdot p(d|ca). \quad (8)$$

where λ_d is proportional to the length of the document $n(d)$.

Interested readers are encouraged to see the more detailed descriptions of the two language models in [11, 9]. In next section, we use these two models as part of features for learning purpose.

3 The Proposed Method

In this section, we first introduce extracted features to levarage learning to rank. Features are used as the input of learning to rank algorithms. So we make a significant effort to devise the features. In the next step, we present the pseudocode of the LREF algorithm, which takes advantage of RankSVM to learn a ranking function.

3.1 Features in LREF

Inspired by profile-based and document-based models, we bind a candidate and relevant documents together as a logical retrieval document and bind a topic query and the documents that best describe the topic of expertise together as a logical query.

To build a logical query for a topic q comprising terms $t_1, \ldots, t_n$, we first find the documents relevant to the topic. Let $R(q)$ be the set of documents retrieved for query q. To build a logical document for a candidate ca, we retrieve the set of documents related to the candidate and denote it D_{ca}. Futhermore, we can through the representation incorporate most expert finding methods into the framework of learning to rank, even if we just take two of them to carry out experiments. For exmaple, voting for candidates [16] can also be absorbed into our method.

Using the representation, we put the features extracted in two groups. One is based on the profiles of candidates and the other on the individual documents. In the following, we detail features:

Profile-Based Features: In this group, we consider a D_{ca} as a single profile document similar to the candidate model and extract freatures depending on the presumption. That is to say, one profile document belongs to one candidate and one candidate only has one profile document. The profile document of a candidate consists of all the documents that relates to the candidate. In traditional information retrieval field, TF, IDF and TF-IDF are basic and popular features [18, 19]. In the view of a document as simply a set of words, known in the literature as the bag of words model, the exact ordering of the

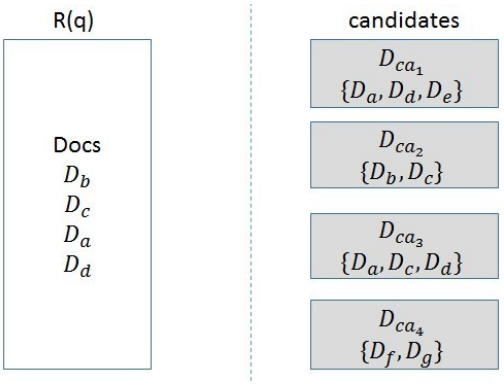


Fig. 1. A simple example from expert finding

terms in a document is ignored but the number of occurrences of each term is material. TF represents *term frequency* and equals to the number of occurrence of term in one document. The *inverse document frequency*, IDF, is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. TF-IDF is the product of term frequency and inverse document frequency. However, different queries may have different numbers of terms, so that we calculate the sum of the number of all the terms in the query. We refer to the feature extraction in the LETOR dataset [21]; nevertheless, the distinction is that we consider each profile document instead of each individual supporting document as a retrieved document. We calculate all features of this group based on the profile documents. Besides, $p(q|\theta_{ca})$ in the candidate model in related work and length of D_{ca} are members of profile-based features, too. Profile-based features are defined as follows:

Table 1. Profile-based features

ID	Feature description
1	$\sum_{t_i \in q \cap D_{ca}} TF(t_i, D_{ca})$
2	$\sum_{t_i \in q} IDF(t_i)$
3	$\sum_{t_i \in q \cap D_{ca}} TF(t_i, D_{ca}) \cdot IDF(t_i)$
4	$LENTH(D_{ca})$
5	$p(q\|\theta_{ca})$

Except for the fifth feature $p(q|\theta_{ca})$, others are all raw information between the query and the candidate and are easy to get in experiments. Feature $p(q|\theta_{ca})$ is borrowed from the candidate model, and its computing method can be found in [11].

Document-Based Features: This group is devoted to gain some features based on the supporting documents. We collect the documents from the intersection of $R(q)$ and D_{ca}. That is all the documents not only related to the topic query q but also the candidate ca. Documents without relation with either the topic q or the candidate ca aren't taken into consideration. Features in this group are on the individual document level, just like document-based models. We collect corresponding features TF, IDF and TF-IDF. However, in most cases there is not only one document related to q and ca. We take the sum of the length of all related documents as a feature. Unlike the previous group, another feature named the number of common documents are devised. At last, we also need the document language model to extract a feature. Table 2 lists the features.

Table 2. Document-based features

ID	Feature description
6	$NUM(D_{ca} \cap R(q))$
7	$\sum_{t_i \in q \cap (D_{ca} \cap R(q))} TF(t_i, D_{ca} \cap R(q))$
8	$\sum_{t_i \in q} IDF(t_i)$
9	$\sum_{t_i \in q \cap (D_{ca} \cap R(q))} TF(t_i, D_{ca} \cap R(q)) \cdot IDF(t_i)$
10	$\sum_{d \in (D_{ca} \cap R(q))} LENTH(d)$
11	$p(q\|ca)$

We have illustrated what features we extracted. Features between a topic and a candidate form a feature vector which is essential to LREF detail described in the following section.

3.2 The LREF Algorithm

LREF formulates the generic expert finding problem as a pairwise learning to rank problem, where candidate pairs are constructed based on their relevance scores. For the actual learning to rank algorithm that returns the ranking f, LREF use RankSVM [22], a classic pairwise learning to rank algorithm. RankSVM takes document pairs as instances in learning, and formulates learning to rank as classification.

In more details, let X be a document, $f_W = W \cdot X$ be a linear function, where W denotes a vector of weights and $\cdot$ denotes inner product. Let X_i^1 and X_i^2 be two documents associated with query q_i. The preference relationship $X_i^1 \succ X_i^2$ means that X_i^1 is more relevant than X_i^2 w.r.t.q_i, which can be expressed by $W \cdot \left(X_i^1 - X_i^2\right) > 0$. Otherwise, $X_i^1 \prec X_i^2$ and $W \cdot \left(X_i^1 - X_i^2\right) < 0$. In this way, the document sets with relevance labels can be transformed into a set of preference instances with lables $y_i = +1 \left(X_i^1 \succ X_i^2\right)$ and $y_i = -1 \left(X_i^2 \succ X_i^1\right)$. These preference instances are the training data for RankingSVM. The loss function of RankingSVM can be represented as follows.

$$\min_{w} \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{l} \xi_i$$

subject to

$$y_i\left(W \cdot \left(X_i^1 - X_i^2\right)\right) \geq 1 - \xi_i, \xi_i \geq 0$$

We now summarize the main procedures of LREF and present the pseudocode in Algorithm 1.

Algorithm 1. The LREF Algorithm

Input: A collection of topic queries Q, a collection of candidates Ca, supporting documents D, and a relevance list R between Q and Ca.
Output: Ranking function f.
Method:

1: $F \leftarrow$ ExtractFeatures(Q,Ca,D)
2: $(Q, Ca \times Ca) \leftarrow$ Formulate(Q,Ca,R)
3: $f \leftarrow$ LearningToRank(Q,$Ca \times Ca$,F)

Line 1 extracts a set of features F, including profile-based features and document-based features. Line 2 formulates the expert finding problem as a pairwise ranking problem, where pairs of candidates $Ca \times Ca$ associated with each topic are generated based on their relevances. Line 3 learns a ranking function f using a learning to rank algorithm such as RankSVM.

The extracted features and the picked learning to rank algorithm have been briefly presented. Then LREF is proposed to integrate them together for expert finding and finally outputs a ranking function which will be used to generate a ranking of experts.

4 Experiments

4.1 Experimental Setup

We evaluate the proposed LREF on the UvT Expert collection[1]. The collection is based on the Webwijs ("Webwise") system developed at Tilburg University (UvT) in the Netherlands. Webwijs is a publicly accessible database of UvT employees involved in research or teaching. Webwijs is available in Dutch and English, and this bilinguality has been preserved in the collection. Every Dutch Webwijs page has an English translation. Not all Dutch topics have an English translation, but the reverse is true: the English topics all have a Dutch equivalent.

We use the entire English corpus, and only the titles of topic descriptions. To begin with, necessary preprocessing is performed on the dataset, especially validating and parsing *xml* files. Then, the standard analyzer in lucene is used to

[1] `http://ilk.uvt.nl/uvt-expert-collection/`

tokenize the documents and topics. We evaluate the methods with mean average precision (MAP) [20], which is the official evaluation measure of expert finding task in enterprise track. Precision at position n for query q is

$$P(q)@n = \frac{\{\#\text{relevant candidates in top } n \text{ results}\}}{n}. \tag{9}$$

Average precision for query q is

$$AP(q) = \frac{\sum_n P(q)@n \cdot I\{\text{candidate } n \text{ is relevant}\}}{\{\#\text{relevant candidates}\}}. \tag{10}$$

For all queries Q, we can get

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q). \tag{11}$$

What's more, mean reciprocal rank (MRR) [20] is also picked up as a measure. Mean reciprocal rank is a statistic measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q :

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}. \tag{12}$$

Evaluation scores were computed using the `trec_eval` program[2].

In our experiments, the English topics were partitioned into 3 parts for 3-fold cross validation, where 2 parts were for training and 1 part for testing, and the averaged performance was reported. We used the candidate language model and the document language model as comparison partners. Besides, we wanted to examine the effect of document features, LREF with only profile-based features and with only document-based features were also carried out as comparative test. We expected to see the performance benefits from the basic document features.

4.2 Experiments Results

Table 3 shows the performance comparison under MAP and MRR measures. It is easy to see that our LREF performs better than the comparison partners on the UvT Expert collection. In addition, the average performance of LREF is 0.4013 in terms of MAP, which is better than 0.3746 with candidate model and 0.3876 with document model. With respect to MRR measure, we also find LREF performs better than the two language models, gaining 23.75% and 2.56% respectively. Thus, we can say that LREF benefits from the combination of the features and then gives better performance.

[2] For registered participants,trec_eval is available from the TREC web site `http://trec.nist.gov`

Table 3. performance comparison of different models

approach	MAP	MRR
Candidate Model	0.3746	0.7637
Document Model	0.3876	0.9215
LREF	**0.4013**	**0.9451**

LREF is a hybrid method in which the profile-based features and the document-based features complement each other. In fact, document-based features can supply useful information on the document level which are beneficial to the people possessing a wide variety of expertise. However, a person who concentrates on only a few topics can benefit from the profile-based features. As we all know, learning to rank is effective in automatically tuning parameters, in combining multiple pieces of evidence, and in avoiding over-fitting. Therefore, we can infer that LREF learns the optimal way of combining these two group features. Then the profile-based features and the document-based features are appropriately weighted in the ranking function generated by LREF. Performance better than the candidate model and the document model demonstrates the effectiveness of LREF.

Table 4. effectiveness of document features for language models

approach	MAP	MRR
Candidate Model	0.3746	0.7637
LREF with only profile-based features	0.3863	0.9228
Document Model	0.3876	0.9215
LREF with only document-based features	0.3952	0.9234

In order to examine the effectiveness of document features for language models, we carry out LREFs with only profile-based features and with only document-based features to compare with the corresponding language models. The results in Table 4 shows that expert finding can benefit from the incorportion of the document features. By comparing the candidate model with LREF with only profile-based features, it's easy to deduce that document features devote to the performance improvement, gaining 3.12% in terms of MAP, 20.8% in terms of MRR. LREF with only document-based features gains 2% improvement than the document model in terms of MAP and has no marked improvement in terms of MRR. Thus we infer that the document features offer valuable efforts to the estimation. Even though TF, IDF, TF-IDF are simple, the RankSVM algorithm takes advantage of the combination of these features to generate a better ranking.

To sum things up, the presented results imply that our LREF method can get more useful information from the dataset than the language models, which helps to escalate the performance of expert finding. It is important and beneficial to study the effect of multiple document features for expert finding.

5 Conclusions and Further Work

In this paper we have presented LREF, the first attempt that effectively adapts learning to rank to expert finding systems. Experiments on the benchmark dataset have shown the promise of the approach. Through the experimetns, it is proved that document features are effective for the language models and collecting document features and language models together lead to performance improvement. LREF can also be considered as a mixed method that incorporate the language models and document features.

Several directions of improvement can be followed in the future. On one hand, we can extract more classic document features, such as BM25 and LMIR and so on. In this way we can leverage many fruits in information retrieval to extend LREF. On the other hand, we can also subsume other expert finding methods into LREF. For example, some methods consider the use of window-based co-occurrence [13] and some methods take advantage of pseudo-relevance feedback [15]. In addition, there are many other choices of learning to rank algorithms, for example, RankBoost, AdaRank and ListNet [21]. The impact on retrieval quality from the other algorithms needs to be further proved while our experiments show initial success with RankSVM. Besides, it is important to consider efficiency as in the case of learning to rank for expert finding.

Acknowledgments. This research is supported by the 863 project of China (2013AA013300), National Natural Science Foundation of China (Grant No. 61375054) and Research Fund for the Doctoral Program of Higher Education of China (Grant No.20100002120018).

References

1. Zellhofer, D.: A permeable expert search strategy approach to multimodal retrieval. In: Proceedings of the 4th Information Interaction in Context Symposium, pp. 62–71. ACM (2012)
2. Mai, X., Ding, G., Wang, J.: Autority aware expert search: Algorithm and system for NSFC. In: Automatic Control and Artificial Intelligence, pp. 585–588. IET (2012)
3. Guy, I., Avraham, U., Carmel, D., Ur, S., Jacovi, M., Ronen, I.: Mining expertise and interests from social media. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 515–526 (2013)
4. Davoodi, E., Kianmehr, K., Afsharchi, M.: A semantic social network-based expert recommender system. Applied Intelligence, 1–13 (2013)
5. Kardan, A., Omidvar, A., Farahmandnia, F.: Expert finding on social network with link analysis approach. In: Electrical Engineering (ICEE), pp. 1–6. IEEE (2011)
6. Yimam-Seid, D., Kobsa, A.: Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach. Journal of Organizational Computing and Electronic Commerce 13(1), 1–24 (2003)
7. Yimam, D.: Expert finding systems for organizations: Domain analysis and the demoir approach. Beyond Knowledge Management: Sharing Expertise (2000)

8. Zhu, J., Huang, X., Song, D., Rüger, S.: Integrating multiple document features in language models for expert finding. Knowledge and Information Systems 23(1), 29–54 (2010)
9. Fang, H., Zhai, C.: Probabilistic models for expert finding. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 418–430. Springer, Heidelberg (2007)
10. Craswell, N., de Vries, A.P., Soboroff, I.: Overview of the TREC 2005 Enterprise Track. In: Trec, vol. 5, p. 199 (2005)
11. Balog, K., Azzopardi, L., De Rijke, M.: Formal models for expert finding in enterprise corpora. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–50. ACM (2006)
12. Fang, H., Zhou, L., Zhai, C.: Language Models for Expert Finding–UIUC TREC 2006 Enterprise Track Experiments. In: TREC (2006)
13. Cao, Y., Liu, J., Bao, S., Li, H.: Research on Expert Search at Enterprise Track of TREC 2005. In: TREC (2005)
14. Lafferty, J., Zhai, C.: Probabilistic relevance models based on document and query generation. In: Language Modeling for Information Retrieval, pp. 1–10. Springer Netherlands (2003)
15. Macdonald, C., Ounis, I.: Using relevance feedback in expert search. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 431–443. Springer, Heidelberg (2007)
16. Macdonald, C., Ounis, I.: Voting for candidates: adapting data fusion techniques for an expert search task. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 387–396. ACM (2006)
17. Voorhees, E.M., Harman, D. (eds.): Overview of the Fifth Text REtrieval Conference (TREC1-9). NIST Special Publications (2001), `http://trec.nist.gov/pubs.html`
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
19. Büttcher, S., Charles, C., Gordon, V.C.: Information retrieval: Implementing and evaluating search engines. The MIT Press (2010)
20. Voorhees, E., Harman, D.K.: TREC: Experiment and evaluation in information retrieval, vol. 63. MIT Press, Cambridge (2005)
21. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
22. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)

Convolution Neural Network for Relation Extraction*

ChunYang Liu[1], WenBo Sun[2], WenHan Chao[2], and WanXiang Che[3]

[1] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing
lcy@isc.org.cn
[2] The Institute of Intelligent Information Processing, Beihang University, Beijing, China
chaowenhan@buaa.edu.cn,
mike891212@gmail.com
[3] The Institute of Social Computing and Information Retreival, Harbin Institute of Technology
car@ir.hit.edu.cn

Abstract. Deep Neural Network has been applied to many Natural Language Processing tasks. Instead of building hand-craft features, DNN builds features by automatic learning, fitting different domains well. In this paper, we propose a novel convolution network, incorporating lexical features, applied to Relation Extraction. Since many current deep neural networks use word embedding by word table, which, however, neglects semantic meaning among words, we import a new coding method, which coding input words by synonym dictionary to integrate semantic knowledge into the neural network. We compared our Convolution Neural Network (CNN) on relation extraction with the state-of-art tree kernel approach, including Typed Dependency Path Kernel and Shortest Dependency Path Kernel and Context-Sensitive tree kernel, resulting in a 9% improvement competitive performance on ACE2005 data set. Also, we compared the synonym coding with the one-hot coding, and our approach got 1.6% improvement. Moreover, we also tried other coding method, such as hypernym coding, and give some discussion according the result.

Keywords: Relation Extraction, Convolution Network, Word Embedding, Deep Learning.

1 Introduction

Relation extraction focuses on semantic relations between two named entities in natural language text. Tree Kernel based method is a classic method for relation extraction, in which it need to parse the sentence to build tree kernel. However, parsing has very high complexity and it is hard to build a correct parse tree for a long sentence. In Addition, kernel method needs hand-engineering features. As for different tasks, we need to construct different kernel functions. Regarding these, an architecture which is not based on parse tree and can learn features is needed. Deep Neural Network architecture can achieve both two goals above.

* This research was supported by Research Fund for the Doctoral Program for Higher Education of China (New teacher Fund), Contract No. 20101102120016.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 231–242, 2013.

Recently, Deep Neural Network models have been applied to many NLP tasks, such as POS Tagging, Name Entity Recognition, Semantic Role Labeling and Sentiment Analysis. Collobert et al. developed the SENNA system that shares representation across task. The SENNA system integrates POS, NER, Language Model and SRL, which are all sequence labeling tasks. To learn feature vectors automatically, previous research usually use embedded representation of word as input layer of CNN. But this method neglects semantic meaning of words, regarding them unique words separately. In this work, we describe a new representation of word. We use a unique code to represent words with same semantic meaning, called synonym coding.

We propose a novel CNN architecture with synonym coding for relation extraction. In experiments, we compared our CNN architecture with the state of art Kernel methods. We also provide the performance of CNN, coding with word list. Result shows CNN with synonym coding outperformed word list coding by 1.6% on ACE 2005 dataset, and CNN architecture outperformed kernel method by nearly 15%, which is a significant improvement compared to kernel methods.

The rest of the article is as follows. First, we describe related work about CNN on natural language processing. We then detail our CNN architecture and synonym coding, followed by experiments that evaluate our method. Finally, we conclude with discussion of future work.

2 Related Work

Early approaches are usually based on patterns and rules, expressed by regular expression. The pattern methods assume all sentences in the same relation type sharing similar linguistic context. But in terms of wide variety of natural language forms in which a given relation may be expressed by multi ways, including syntactic, morphological and lexical variations. Pattern based methods cannot cover all language forms [1]. This causes very low recall value.

Several researchers have already attempted to build machine learning approaches for relation extraction. [2] presented a report of feature vectors based relation classification. The authors annotate words with rich features, including:

- Part-of-Speech
- The relation arguments
- The entity mention level
- The entity types
- Parse tree

In addition, they proposed to take advantage of parse tree without its structure information. However, entities usually have a long distance in relation extraction. When building parse tree, long range dependence will involve many unrelated words, which will sparse feature space. Parse tree structure provides grammar information of words. To fully using this information, [3] presented tree kernel method to consider structure of parse tree, and got slightly progress compared with approaches based on feature

vectors. [4] used dependency path, removing many syntactic nodes which have indirect dependency relation with target entities. This work produced a slightly improvement then that used full parse tree. But building parse tree and dependency tree is a very time-consuming processing, not only when training, but predicting.

Moreover, kernel method needs hard coding of features. To self-adapt different tasks, researchers import embedded word representation to NLP and obtain some success.

[5] described a lookup table processing and stack it on a multilayer perceptron for building language model, resulting in a 20% improvement on perplexity. Language model is an important build block of machine translation and speech recognition systems, so this improvement advanced the tasks involving language model. [6] proposed neural network architecture for machine translation, resulting in enhancing 2 point of BLEU score. In addition, [7] presented a unity CNN for basic NLP tasks and SLR, got the state-of-the-art performance. All methods above use a lookup table layer to learn words' feature by back-propagation. The input of lookup table is word indexes which are provided from a wordlist. Each word has a unique index, which means even words with similar semantic meaning are resolved separately, ignoring semantic information.

As for compositive applications, [8] trained a deep architecture, auto-encoder, for sentiment classifier, and surpassed the state-of-the-art. Auto-encoder [9] is one of the first processing of deep learning architecture. The goal of an auto-encoder is finding a best representation of input. For sentiment classifier, inputs are words, whose semantic meaning exactly indicates their sentiments. But for relation extraction, different words play unequal roles to identify a relation type. Some words are indicator of relation types, some are unrelated with relation. So, although auto-encoder for words can represent well in many domains, it can't provide enough specific information for relation extraction task, such as grammatical structure of sentences.

3 Convolution Network Architecture

Previous research on relation extraction focuses on how grammatical relation, such as parse tree, expresses semantic relation. Undeniably, parse tree and dependency tree performs well with large number of empirical features. For semantic information, WordNet provides us significant improvement on performance of relation extraction system. But building parse tree is a time consuming task. Moreover, hand-built features are rigid considering context.

Ideally, we want to build a relation extraction system without a time consuming parse tree. Meanwhile, the system should consider of grammatical structures.Also, we want to avoid hand-built features. Hence, we prefer to find an automatic weight learning approach to avoid time-consuming feature generation. We found that a deep neural network achieve both two goals.

To including grammatical information, we proposed convolution neural network (CNN) [10]. In image processing tasks, convolution is used to extract features over blocks. In language processing, CNN can be used to evaluate grammatical relations

between abutting words that probably constitute a phrase. So, CNN can replaces parse tree to provide grammatical information.

3.1 Word Embedding

Word embedding is a method to project words to vectors. The first step of word embedding is using One-hot coding. One-hot coding in NLP is a coding method that transfer word index to a binary code whose size is equal to size of wordlist for the convenience of computing. In one-hot code, only on position of index coded to 1, other positions are all 0.

One-hot coding method is formalized below:

Define word dictionary is D, the i^{th} word in D is w_i. Input sentence is S.

We construct an indice function to index words in the input sentence. s(i) refer to the i^{th} word in the input sentence. Given the notation above, we define one-hot coding of a word as an identity function:

$$C(w) = \{1_{w=w_i}\}_{i\in[0,\#D]} \tag{1}$$

Equation (1) means if and only if i^{th} word in D is equal to input word, element is 1. The other elements are all 0. Synonym coding uses a synonym list instead of wordlist. So, in synonym code, D_s represents the synonym dictionary. The i^{th} synonym cluster is noted by Syn_i. Synonym code is a vector of $\#D_s$ dimension, defined as:

$$SC(w) = (1_{w\in Syn_i})_{i\in[0,\#D_s]} \tag{2}$$

Equation (1) means if and only if i^{th} word in D is equal to input word, element is 1.

Although we have synonym list, the list can't hold all words. So if the input sentence has words beyond D_s. We add the words toD_s and each word composite a seperate synonym cluster. Since we have notations above, an input sentence of n words can be mapped into one-hot coding.

$$S = \{SC(s(i))\}_{i\in[1,n]} \tag{3}$$

3.2 Basic Architecture

The type of neural network we employ here is Convolution Neural Network[10]. Convolution Network has been used in one unity NLP architecture. The neural network concluded into 3 main building blocks:

- Input layer
- Convolution Layer
- Classic Neural Network Layers

Before inputting the sentence to the network, sentences will be mapped into synonym code. In terms of variant of sentences length, we define a window around P_1, P_2, whose size is noted as *wsz*. All words in this window will be coded. And the

distance of P_1, P_2 is noted as b. If b is larger than *wsz*, we set P_1 to left-most, and set value on the right-most position to B. If b is smaller than *wsz*, empty position will distribute around P_1, P_2 on average, and set those position to a sign X. Both B and X will be added to synonym list, participating in coding processing.

Input Layer

For the purpose that words weights can be learnt by back-propagation, we implement a Lookup Table Layer [11] as the input layer of CNN.

In lookup table layer$LT_W(\cdot)$, each word *w* will be mapped into a d-dimension space:

$$\mathrm{LT_W(s(i))} = \mathrm{W} \cdot \mathrm{SC(s(i))} = \mathrm{W_i} \tag{4}$$

where $W \in R^{d \times \#D_S}$is a matrix of parameters to belearnt.$W_i \in R^d$ is the i^{th} column of W. and d is the word feature vector size to be chosen by the user. In the first layer of our architecture an input sentence of n words is transformed into a series of vectors by applying lookup table to each word.

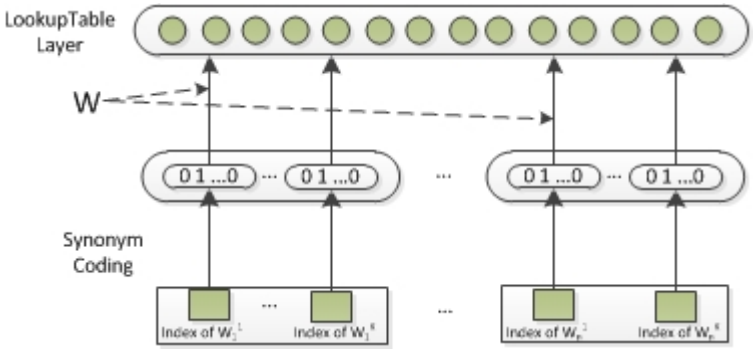


Fig. 1. Lookup Table Layer and Synonym Coding

In relation extraction, the mention level and type of entity are necessary information for classification. So in our experiments, we added Name Entity Type List and Mention Level List to code the name entity mention level and type. Two lists above will generate two new features.

When a word is decomposed into K features, it can be represented as a tuple

$$SC'(s(i)) = \{SC_1(s(i)), SC_2(s(i)), \ldots SC_K(s(i))\} \tag{5}$$

where $SC'(s(i)) \in D^1 \times D^2 \times \ldots \times D^K$, D^K is the dictionary for the k^{th} feature. Each feature will be mapped into a new vector space, so each feature need coding separately. To associate multi features, lookup table for each feature is defined as$LT_{W^k}(\cdot)$, with the parameters$W^k \in R^{d^k \times \#D^k}$where d^k is a user-specified vector size. A word s(i) is then mapped into a $\mathrm{d} = \sum_k d^k$ dimension vector by concatenating all lookup table outputs:

$$LT_{W^1,\ldots,W^K}(s(i))^T = \{LT_{W^1}(s(i)), LT_{W^2}(s(i)), \ldots, LT_{W^K}(s(i))\} \quad (6)$$

Convolution Layer

A convolution layer is typically used in a convolution network for vision tasks. When one want to get a more abstract feature, it's reasonable to ignore some inputs and select a subsequence and compute global weights.

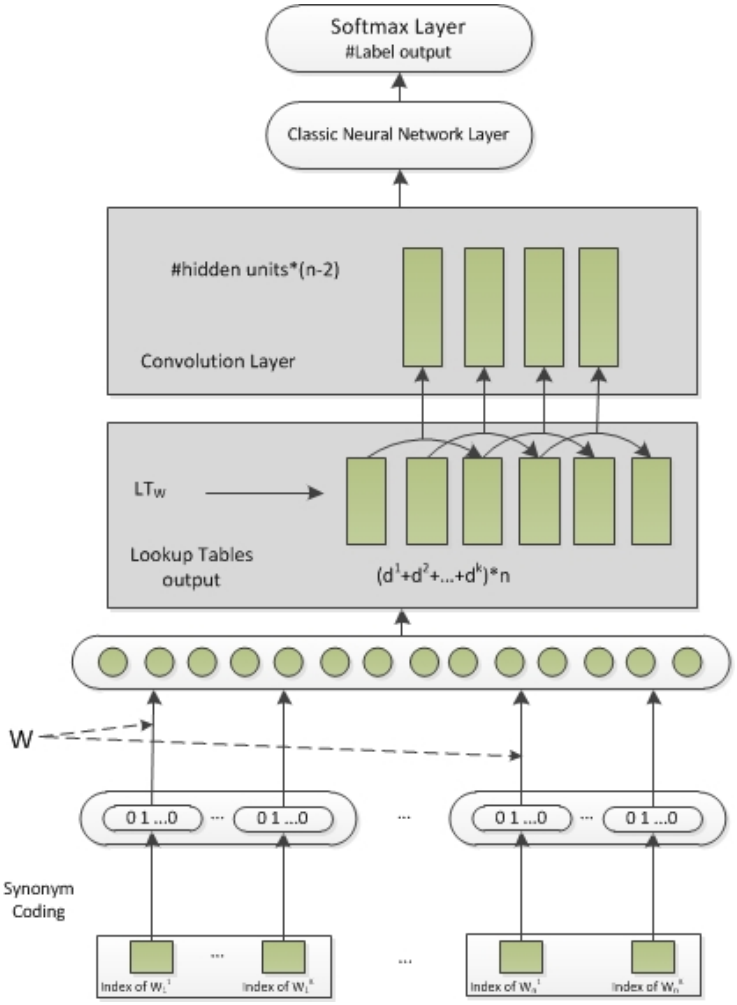


Fig. 2. Deep Neural Network Architecture

A typical convolution usually applied on 2D data, which can be visualized to an image. But in case of our sequential input, convolution layer of our CNN have a sequential kernel, which is an exception of typical convolution kernel. A convolution kernel maps a subsequence to a new vector space. The width of the kernel decides the

dimension of new vector space and step decides number of subsequence involved in convolution. A convolution computation on one subsequence is:

$$\text{Convolution}\left(s_{i,i+width-1}\right) = \sum_{j=1}^{width} L_j \cdot s_{i+j} \tag{7}$$

where $L_j \in R^{n \times d}$, $s_j \in R^d$. So the result convolution is a matrix of R^n.

Concatenating convolutions on all subsequence referring of lookup table layer's outputs, referring to (6), we'll get $\text{Convolution}\left(LT_W(s)\right)$

$$\left\{\text{Convolution}\left(LT_W(s)_{i,i+width-1}\right)\right\}_{i \in \{1,1+step,\ldots,wsz\}} \tag{8}$$

The dimension of each convolution result is number of hidden units. After convolution on whole sequence, it results in a matrix of$R^{hu \times (wsz-step)}$. We then add to the architecture a layer, which captures the most relevant features over the window by feeding CNN layers into a Max Layer, which takes the maximum over a row in (8) for each of the n_{hu} outputs.

3.3 General Deep Neural Network Architecture

Combining the lookup table layer and convolution layer, we'll get basic convolution network. The output of lookup table layer is the input of convolution layer. A max layer, which is applied to the output of convolution layer, reducing output dimension to number of hidden unit, is stack on the convolution layer. To classify relation types, we add a softmax layer to be output layer. Softmax Layer:

$$h_i(z) = \frac{\exp z_i}{\sum_j \exp z_j} \tag{9}$$

Where *i* indicates i^{th} output unit, and j refers to j^{th} input of softmax layer.

$$\sum_i h_i(z) = 1 \tag{10}$$

(9) and (10) allows us to interpret outputs as probabilities for each relation type prediction. The size of outputs is the number of classes of relation extraction.

Between convolution layer and output layer, some classic neural network layers can be added to get more abstract feature. The whole architecture can be summarized in Figure 2.

The whole network trained by a normal stochastic gradient decent with negative log-likelihood criterion.

4 Experiments

In this section, we compare the performance of our new CNN architecture with the performance of tree kernel method. To prove effectiveness of synonym coding, we also compare CNN with synonym coding input with that without synonym coding.

4.1 Date Set

The Automatic Content Extraction (ACE) Evaluation covers the Relation Detection track. ACE 2005 [12] dataset consists of 599 documents which are related with news, speech and email. As ACE 2005 defined, all relations are annotated to 7 major types and 19 subtypes. Our training set covers 6 major types: ART (686 instances), GEN-AFF (1280 instances), ORG-AFF (395 instances), PART-WHOLE (1009 instances), PER-SOC (465 instances), PHYS (469 instances), and 18 subtypes. The Metonymy type includes very small amount of instances, this is not enough to evaluate performance on classification for this type. So we abandoned it.

ACE training set provides entity mention level and entity type features, we extract them and generate their feature list. We also add Part-of-Speech feature generated by Stanford POS tagger [15]. So we finally have five feature lists: word list (after stemming), POS list, mention level list, entity major type list, entity subtype list.

4.2 Experiments Setup

As for the tree kernel classifier, we use Stanford parser [13] to obtain parse tree and dependency tree, and implemented all reported tree kernels by LibSVM [19].

For all kernels involved, we use same SVM parameters. We have conducted a 5-fold cross validation. In addition, we also extract 30% out of all training data to be test set.

4.3 Result

In the results, we report usual evaluation measure, Precision and Recall, comparing the performance with kernel-based method on unbalanced corpus.

Table 1. Precision, recall, and F-measure for 5-fold cross validation and test set. Here SPK denotes the Shortest Path Kernel [16], TDK denotes the Typed Dependency Kernel [17], CK denotes the Context-Sensitive Tree Kernel[18]

	5-fold cross validation			Test set		
	Precision	Recall	F1	Precision	Recall	F1
CNN	**0.868**	**0.875**	**0.872**	**0.837**	**0.839**	**0.838**
SPK	0.821	0.472	0.599	0.803	0.455	0.581
CK	0.812	0.658	0.727	0.796	0.641	0.710
DTK	0.822	0.702	0.758	0.811	0.688	0.744

The CNN outperforms the tree kernel methods by 9 points on F-measure, which is a significant improvement.

The state-of-the-art tree kernel gives a 74% F-measure, this is to say, kernel function is a good method to represent relation. But construction of kernel function need experiential coding for features. This cannot give the kernel function the best representation on given task. The performance of CNN on subtypes, tested by 5-fold cross validation, is showed in Table 2.

Table 2. Theperformance of CNN, SPK and TDK on subtypes

	Precision	Recall	F1
CNN	0.748	**0.748**	**0.748**
SPK	0.743	0.303	0.430
CK	**0.759**	0.549	0.637
DTK	0.755	0.599	0.668

Table 2 shows CNN has poor performance on subtypes, because 5 types in subtypes only have less than 20 instances. Obviously, this is not enough to train a classifier. The F-measures on 3 subtypes are only about 0.1. But when we evaluate performance on subtypes by F1, which considers the amount of different types, avoiding unbalanced dataset pulling down the performance, the F1 achieved **74.8%**, better than **66.8%** of DTK. This shows our architecture is effective on relation extraction.

ACE data set do not has enough words to train a word embedding layer for better semantic information, so we add synonym list in WordNet to provide more semantic information for words.

To generate synonym coding, we use Synonym List in WordNet 2.1 [14]. There are 8049 stemmed words in wordlist conducted from training set. After coding, the size of list reduces to 6741 words.

We compared CNN with synonym coding with CNN without that, performance showed in Table 3.

This result shows Synonym coding improve the performance by 1.6% than CNN without Synonym coding. The synonym coding method is proved to be effective here.

Table 3. Performance of CNN with synonym coding and that without synonym coding

	Precision	Recall	F1
Synonym	**0.837**	**0.839**	**0.838**
Non-Synonym	0.837	0.813	0.825

Variant of window size and feature dimension may influent performance of CNN, so we conducted another experiments on different wsz and d^k . The results are showed in table 4.

Table 4. Performance on different window size and different feature dimension

	wsz=7			wsz=15			wsz=21		
d^k	P	R	F1	P	R	F1	P	R	F1
15	0.769	0.762	0.765	0.823	0.818	0.821	0.835	0.837	0.836
50	0.766	0.785	0.767	0.825	0.822	0.824	0.837	0.839	0.838
100	0.769	0.756	0.762	0.816	0.807	0.812	0.807	0.802	0.804

On variant window size, result shows that larger *wsz* will get better performance. This appearance showed difference against dependency kernel, whose effect reduces unrelated words. In the dependency tree kernels reported in the first experiment, most of the node number on dependency path is around 5 or 6, smaller than the minimum *wsz* in CNN. This phenomenon can be explained by that words in CNN don't provide any grammar structure information, leading to no initial weight of words are given. But in tree kernel methods, words on dependency path indicate these words are more important than other words. So CNN needs more other words to learn which words are more effective. Besides, we also observed that the maximum feature dimension didn't give a better performance, but the middle one.

We also used other coding table except synonym table, such as, hypernym table from WordNet. We use three hierarchies to integrate words into their hypernym. The result shows in table 5.

Table 5. Performance using synonym table and hypernym table

	Precision	Recall	F1
Synonym	0.837	0.839	0.838
Hypernym-1	0.828	0.822	0.824
Hypernym-2	0.821	0.815	0.817
Hypernym-3	0.819	0.815	0.817

An interesting phenomenon is that as the semantic hierarchy going up, performance of CNN gets no enhance. So, we tried to cluster word feature vectors and got some interesting results.

Table 6. Sample of word clusters

Cluster36	Cluster88
Neichangshan, F-15e, RigobertoTiglao, Hezbollah	George W. Bush, High Court, Haditha, Al Anbar

In these three clusters, we can observe three different topics, which is showed by words' semantic meaning obviously. Here we spread out some words in cluster. In cluster1, most of words are about politics. And In cluster2, weapons are listed. In training processing, lookup table layer learn words vectors by back-propagation. After training, the words are clustered by their semantic meaning, like hypernym. But these clusters are more adaptive to the relation extraction task. Adding hypernym could be regarded as another hard coding comparing with self-learnt clusters. So Adding hypernym leads to negative effects on classification. Because of the limited corpus size, we didn't get more specific clusters, but we observe the trend of self-organization on semantic meaning.

In conclusion, CNN with Synonym coding give a better performance than the state-of-the art kernel method.

5 Discussion and Future Work

From the experiments above, we find that CNN have better performance on relation extraction. Synonym coding method is also an effective coding method to improve overall performance. Actually, only 16% words in wordlist are coded to synonym after synonym coding. In this trend, if we use a larger synonym coding list, which can code more words, the performance may improve.

Moreover, on a conceptual prospective, synonym coding is just a coding method to reduce input space before project words to vectors. Other coding method or word selection method can reduce input space too. For example, if we only consider the words on dependency path, which compose the advantage of dependency tree kernel with CNN architecture. Our future work will focus on finding more effective coding method and word selection.

Although CNN give us a better resolution on relation extraction, there are some problems about CNN architecture we should notice. The architecture doesn't give us a guide how to choose the feature dimension. Here in our method, a convolution layer is included. We also tried using linear layer to find the divergence with convolution layer. Result shows convolution layer products better performance, but we cannot find a way to explain this divergence. So for the future work, we will try to find some theoretical method to explain the differences when using variant layer and find a guild line to help us determine the feature dimension.

References

1. Grishman, R.: Information Extraction: Capabilities and Challenges. Lecture Notes of 2012 International Winter School in Language and Speech Technologies, Rovirai Virgili University (2012)
2. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In: The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 178–181 (2004)

3. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. Journal Machine Learning Research 3, 1083–1106 (2003)
4. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pp. 423–430 (2004)
5. Collobert, R., Weston, J.: Fast Semantic Extraction Using a Novel Neural Network Architecture. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 560–567 (2008)
6. Schwenk, H., Rousseau, A., Attik, M.: Large, pruned or continuous space language models on a gpu-forstatistical machine translation. In: Workshop on the Future of Language Modeling for HLT (2012)
7. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: ICML 2008 (2008)
8. Grolot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach.In ICML 2011 (2011)
9. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. Biological Cybernetics 59, 291–294 (1988)
10. LeCun, Y., Bengio, Y.: Convolutional Networks for Images, Speech, and Time-Series. The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
11. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A Neural Probabilistic Language Model. Journal of Machine Learning Research 3, 1137–1155 (2003)
12. Walker, C., Strassel, S., Medero, J., Maeda, K.: ACE 2005 Multilingual Training Corpus. Linguistic Data Consortium, Philadelphia (2006)
13. Klein, D., Manning, C.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)
14. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
15. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL, pp. 252–259 (2003)
16. Bunescu, R., Mooney, R.: Proceedings of the 19th Conference on Neural Information Processing Systems (NIPS), Vancouver, BC (2005)
17. Reichartz, F., Korte, H., Paass, G.: Semantic Relation Extraction with Kernels Over Typed Dependency Trees. In: KDD 2010, Washington, DC (2010)
18. Zhou, G., Zhang, M., Ji, D., Zhu, Q.: Tree Kernel-based Relation Extractionwith Context-Sensitive Structured Parse Tree Information. In: EMNLP 2010, Prague, pp. 728–736 (2007)
19. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)

Extracting Fuzzy Rules from Hierarchical Heterogeneous Neural Networks for Cardiovascular Diseases Diagnosis

YuanLian Cui and MingChui Dong

Faculty of Science and Technology, University of Macau, Avenida Padre Tomas Pereira, Taipa, MACAU SAR, China
maomaoqiupanda@gmail.com, charley_dong@hotmail.com

Abstract. Although hierarchical fuzzy neural networks (FNNs) perform with high accuracy in medical diagnosis systems, their popularity is held back from well-known disadvantage of not providing explanation. This paper presents a novel rule extraction approach to extract accurate and comprehensible fuzzy IF-THEN rules via genetic algorithm (GA) from hierarchical heterogeneous FNNs (HHFNNs). When each sub-FNNs is constructed and trained, entire HHFNNs are constructed and trained jointly through integrating all trained sub-FNNs. The proposed rule extraction approach is used to extract rule set from each concerned sub-FNNs, all extracted rule sets are then combined as one set to provide automatically exclusive explanation to diagnostic conclusion when IF part contains input features and THEN part contains diagnostic conclusions. Experimental study on diagnosing three most common and important cardiovascular diseases using hospital site-measured data demonstrates that such proposed approach exhibits satisfactory explanation capability without concerning inner structures of HHFNNs.

Keywords: hierarchical heterogeneous fuzzy neural networks, fuzzy logic, rule extraction, genetic algorithm, fuzzy IF-THEN rules.

1 Introduction

Hierarchical neural networks (NNs) have been successfully used and shown merits in many areas, such as multiple fault diagnosis, medical prognosis and pattern classification etc. [1-3]. However, NNs regarded as opaque models represent knowledge implicitly in their hidden nodes and link weight matrix, which leads to their rather low degree of comprehension. Therefore to redress the opaqueness of NNs, rule extraction approaches applied to trained NNs have attracted much attention. However, some of them extract rules from trained NNs with discrete or linguistic input features or disjoint intervals through pre-partitioning continuous ones [4, 5]; some of them are developed and applied for just one particular type of NNs [6, 7].

Fuzzy logic can deal with uncertainty or ambiguous data. Fusing fuzzy logic with NNs in various manners has been researched [8-12]. Methods of extracting fuzzy IF-THEN rules from trained fuzzy NNs (FNNs) are proposed in [13, 14].

This paper proposes an approach of extracting fuzzy IF-THEN rules using genetic algorithm (GA) for hierarchical heterogeneous FNNs (HHFNNs). GA is a search algorithm that is computationally simple yet powerful and proven to be robust for solving complex problems with large search space [15]. In this paper, GA is used to

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 243–249, 2013.

search compact fuzzy rule set. The proposed rule extraction approach is a pedagogical method which can be used to any types of NNs and is independent of NNs' inner structure and training algorithm. Quantitative features are converted to fuzzy membership values as input features of sub-FNNs. All sub-FNNs are trained and integrated to construct entire HHFNNs. Fuzzy IF-THEN rules are extracted from each concerned sub-FNNs to form corresponding fuzzy rule set. Afterwards combine all extracted fuzzy rule sets by generating final rule set with rules where IF part contains input features and THEN part contains diagnostic conclusions. Such final rule set is used to provide exclusive explanation to the diagnostic conclusions of entire HHFNNs.

The proposed method is verified on diagnosing three most common and important cardiovascular diseases (CVDs), namely coronary heart disease (CHD), hypertension (HT), and hyperlipaemia (HL). Hospital site-sampled hundreds medical records are used as dataset.

2 Methodology

2.1 HHFNNs

The system overview of proposed HHFNNs is illustrated in Fig. 1. Such proposed HHFNNs are developed for diagnosing diseases with features that can be categorized into different groups (Group_1, 2, …, G). Features in each group are used as input features of corresponding sub-FNNs to conduct their result. The NNs can be any type such as back-propagation, learning vector quantization, probabilistic, radial basis function etc. Different types of NNs might appear different efficiencies or performances to different input features, consequently integrating different types of sub NNs forms hierarchical heterogeneous NNs with superior advantages [16].

Here an example is given to indicate the working principle of proposed HHFNNs. Features of Group_1 (x_1^1, x_2^1,…, x_m^1) are treated as input parameters of FNNs_1 to get

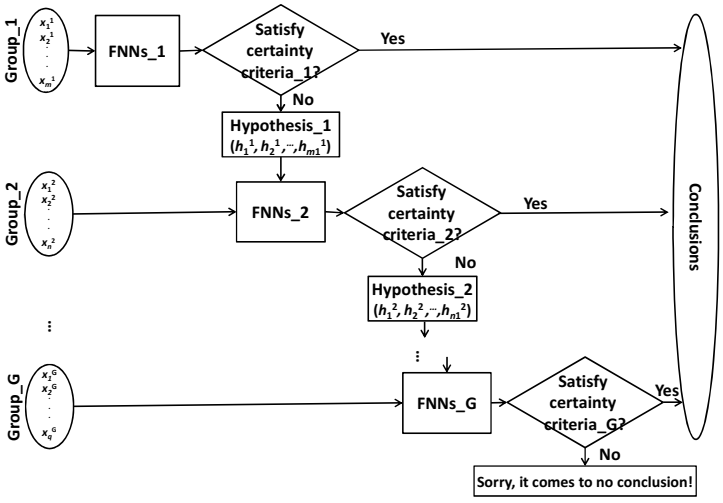


Fig. 1. System overview of proposed HHFNNs

outputs: if the outputs satisfy the pre-set criteria, outputs are accepted as conclusions of entire HHFNNs; otherwise, the outputs are marked as hypothesis_1 (h_1^1, h_2^1,..., h_{m1}^1) which will work jointly with features of Group_2 (x_1^2, x_2^2,..., x_m^2) and work out the conclusions of FNNs_2. Likewise, layer by layer, the final conclusions are generated by HHFNNs at certain layer. In the worst case, after carrying out previous procedure until the last FNNs_G suppose the conclusions still cannot satisfy the termination criteria, just same as physicians do not work out conclusions sometimes in hospital, the HHFNNs have to display "Sorry, it comes to no conclusion!" as the final result.

2.2 Process of Providing Explanation to Conclusion of HHFNNs

As for process of providing explanation to the conclusion directly generated by FNNs_1, it can be viewed as rule extraction from single type of NNs to give exclusive explanation. The concrete procedures of fuzzy rule extraction from single type of NNs are represented in section 2.3. As for more generic conclusion conducted by FNNs_*i* (*i* = 2, 3, …, G) with inputs Database_*i* which consists of Group_*i* and Hypothesis_*i*-1, the exclusive explanation provided for it can be generated through a comparatively complicated process as illustrated in Fig. 2. Just consider related output branch of trained FNNs_*i*, either conclusion or hypothesis.

1. Fuzzy rule extraction method using GA is implemented on trained FNNs_*i* to obtain fuzzy rule set^i.
2. Fuzzy rule set^k (k = i-1, …, 1) must be extracted from FNNs_k, with which the hypothesis_k part of inputs of FNNs_k+1 are associated.
3. Combine all extracted fuzzy rules to get fuzzy rule set^j (j = i+1) where IF part consists of input features Group_*i* and THEN part contains the conclusion needed to be explained. For example, when i = 2 combine fuzzy rules in set^1 (IF x_1^1 is A and x_2^1 is B, THEN h_1^1) and set^2 (IF h_1^1 and x_1^2 is C, THEN disease_1) and get set^3 (IF x_1^1 is A, x_2^1 is B and x_1^2 is C, THEN disease_1). (A, B and C can be real numbers or fuzzy subsets).
4. Provide explanation with fuzzy rule set^j. If NNs' conclusion to be interpreted is covered by one rule of fuzzy rule set^j, obviously this rule can be used to provide exclusive explanation to the conclusion [17].

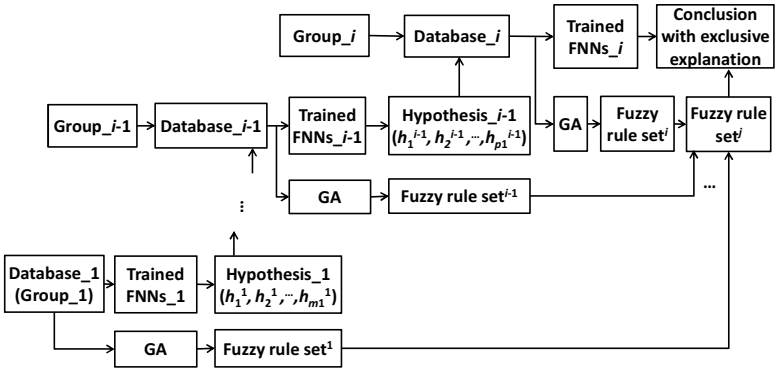


Fig. 2. Function flow chart of fuzzy rule extraction via GA from HHFNNs

2.3 Fuzzy Rule Extraction Approach Using GA

In a previous paper [17], the approach of extracting rules from trained NNs via GA was implemented on three types of NNs and had a good performance, which proved its independence of NNs' architecture and training algorithm. In this paper, adapted algorithm is adapted and implemented on sub-FNNs of HHFNNs. The algorithm of rule extraction from sub-FNNs is elaborated bellow.

1. Convert quantitative features into sets of fuzzy membership values based on fuzzy logic.
   Each quantitative feature can be expressed in terms of membership values corresponding to its linguistic expression which might be: 1 stands for *Very Low*, 2 for *Medium Low*, 3 for *Normal*, 4 for *Medium High*, and 5 for *Very High*. According to the characteristics of input feature distribution, the Gaussian function is used for representing the distribution in value ranges of "*Medium Low*", "*Normal*", "*Medium High*", while Z function and S function denote the distribution in value ranges of "*Very Low*" and "*Very High*" respectively.
2. Construct and train sub-FNNs.
   Use pre-fuzzified and specifically grouped features as input features for that sub-FNNs. The architecture and training algorithm of NNs are not constrained.
3. Extract fuzzy rules via GA from trained sub-FNNs.
   As Fig. 3 shows the flow of GA, to begin with some individuals consisted of multiple genes are generated as an initialization. Then evaluate each individual based on fitness function. Roulette wheel selection method is applied to these individuals based on their fitness. And then crossover and mutation are implemented on individuals. Carry out these genetic operations until the number of generation (ng) reaches the maximum number (N).

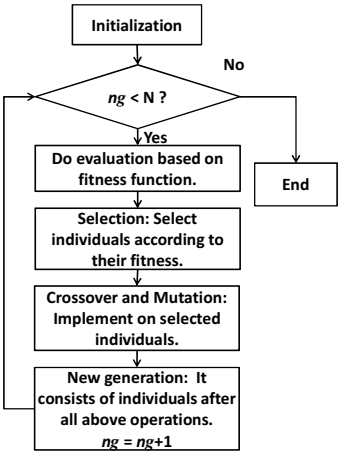


Fig. 3. The flow chart of GA

Each chromosome represents individual IF-THEN rule, which consists of genes coding premises and a single gene coding the conclusion. Fig. 4 shows the designed form of chromosome. The rule may have different number of premises by changing binary flags. The premise can be included in the body of rule only when its flag is set to 1.

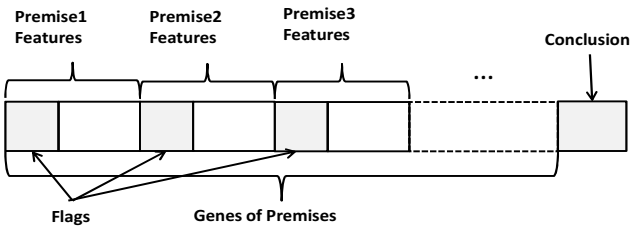


Fig. 4. The designed form of chromosome

The purpose is to generate the fewest rules that classify most examples same as sub-FNNs do. Therefore, the accuracy of the extracted rule set must be maximized; the extracted rule set must have high comprehensibility. We define the following fitness function:

$$Fitness = W_{Acc}\frac{TP}{TP+FP} - W_{Cc}\frac{C}{Max\_C} - W_{Cr}\frac{R}{Max\_R} \quad (1)$$

where the weights W_{Acc} (*Acc* means accuracy), W_{Cc} (*Cc* means chromosome related comprehensibility), and W_{Cr} (*Cr* means rule related comprehensibility) are defined by the users' empirically; *TP* is the number of examples covered by this extracted rule where conclusion predicted by the rule is same as the one generated by NNs; *FP* is the number of examples covered by this extracted rule where conclusion predicted by the rule is different from the one generated by NNs; *C* is the number of premises included in this extracted rule; *Max_C* is the possible number of maximum premises included in this extracted rule; *R* is the number of extracted rules; *Max_R* is the possible number of maximum rules.

3 Test Results

3.1 Dataset

This proposed rule extraction method has been evaluated on dataset established by our team using hospital site-sampled medical records for CVDs diagnosis which includes patients' physiological messages, original sphygmogram data, hemodynamic parameters (HDPs) and doctor's diagnosis results. Such constructed dataset contains 38 parameters including 6 physiological messages and 32 HDPs. Using variance analysis, the relevant categorization and setting priority for discrimination of CHD, HT and HL can be implemented on these parameters; and three groups (4 medical records in sensitive, 12 in auxiliary and 22 in replenish group separately) with different confidence

coefficients are obtained [16]. Correspondingly, proposed HHFNNs have three grouped input features: XGroup_1 = [Age CO FEK PAP]T; XGroup_2 = [Weight SP DP PR SI CI HOV CMBV PP MSP SPR yr]T; XGroup_3 = [SV VPE EWK SWI HOI CMBR MDP MAP CCP BLK AC TPR VER PAWP BV y MHR MRT MST Height Sex PAR]T. Full names of these acronyms can be found in Table 1 in [2].

3.2 Test

Inherited from previous study [16], proposed HHFNNs adapt fuzzy probability NNs for FNNs_1 and FNNs_2, use fuzzy learning vector quantization networks for FNNs_3. The HFFNNs are constructed and trained; the proposed fuzzy rule extraction via GA method is implemented on. Some examples of providing explanation to HHFNNs' conclusion are enumerated in Table 1. Of the three groups, only contributing ones are listed.

Table 1. Some examples of providing explanation to HHFNNs' conclusion

Patient's data	HHFNNs' conclusion	Corresponding IF-THEN rule for explanation
$\mathbf{X}_{Group\_1}$ = [46.3505 7.9193 0.4198 23.5258]^T	non-CHD	IF Age is Medium High, FEK is Normal and PAP is Normal THEN the patient belongs to non-CHD class
[$\mathbf{X}_{Group\_1}$ $\mathbf{X}_{Group\_2}$] = [55 5.6353 0.5280 10.1949 75 120 90 71.2173 42.7907 3.0461 36.3549 370.1931 30 112.1805 1533.9460 4.7374]^T	CHD	IF PAP is Very Low, Weight is Normal, SI is Very Low, CMBV is Normal, PP is Medium Low, SPR is Very High and yr is Very High THEN the patient belongs to CHD class
[$\mathbf{X}_{Group\_1}$ $\mathbf{X}_{Group\_2}$ $\mathbf{X}_{Group\_3}$] = [77.3 6.7204 0.4466 18.0445 70 120 70 63.6086 61.6578 3.9236 31.3584 285.6853 40 90.3421 866.8507 2.1500 105.6074 1.8459 0.2837 65.4209 18.3083 1.1189 68.5044 79.4233 47.3372 0.2182 1.5414 954.6391 205.2107 12.6628 6.0312 2.9136 0.0371 18.8411 27.1877 160 1 139.5928]^T	CHD	IF Age is Very High, HBPa is Medium High, SI is Medium Low, CI is Normal, VPE is Very High, EWK is Normal, HOI is Very High, CMBV is Medium Low, MAP is Normal, TPR is Normal, SPR is Very Low and PAR is Medium Low THEN the patient belongs to CHD class

4 Conclusion

A novel pedagogical rule extraction method is presented for extracting fuzzy IF-THEN rules from HHFNNs via GA. This method is evaluated on database consisted of hospital site-sampled medical records for CVDs diagnosis, which is high dimension

and continuous. Using extracted final fuzzy rule set, the exclusive explanation to HHFNNs' final diagnostic conclusion in CVDs diagnosis system is generated automatically, thus solve the opaqueness of NNs in e-health system efficiently.

Acknowledgements. This work was supported in part by the Research Committee of University of Macau under Grant MYRG184 (Y1-L3)-FST11-DMC, and in part by the Science and Technology Development Fund (FDCT) of Macau S.A.R under Grant 018/2009/A1.

References

1. Calado, J.M.F., Costa, J.M.G.: A Hierarchical Fuzzy Neural Network Approach for Multiple Fault Diagnosis. In: Proc. UKACC Int. Conf. Control 1998 (Conf. Publ. No. 455), vol. 2, pp. 1498–1503 (1998)
2. Shi, J., Sekar, B.D., Dong, M.C., Lei, W.K.: Fuzzy Neural Networks to Detect Cardiovascular Diseases Hierarchically. In: Proc. 10th IEEE Int. Conf. Comput. Inf. Technol., CIT 2010, pp. 703–708 (2010)
3. Fay, R., Schwenker, F., Thiel, C., Palm, G.: Hierarchical Neural Networks Utilising Dempster-Shafer Evidence Theory. In: Schwenker, F., Marinai, S. (eds.) ANNPR 2006. LNCS (LNAI), vol. 4087, pp. 198–209. Springer, Heidelberg (2006)
4. Duch, W., Adamczak, R., Grabczewski, K.: Extraction of Logical Rules from Neural Networks. Neural Processing Letters 7, 211–219 (1998)
5. Chorowski, J., Zurada, J.M.: Extracting Rules from Neural Networks as Decision Diagrams. IEEE Transactions on Neural Networks 22(12), 2435–2446 (2011)
6. Gupta, A., Park, S., Lam, S.M.: Generalized Analytic Rule Extraction for Feedforward Neural Networks. IEEE Transactions on Knowledge and Data Engineering 11(6), 985–991 (1999)
7. Fu, X.J., Wang, L.P.: Linguistic Rule Extraction from a Simplified RBF Neural Network. Computational Statistics 16(3), 361–372 (2001)
8. Pal, S.K., Mitra, S.: Multilayer Perceptron, Fuzzy Sets, and Classification. IEEE Transactions on Neural Networks 3(5), 683–697 (1992)
9. Mitra, S.: Fuzzy MLP Based Expert System for Medical Diagnosis. Fuzzy Sets and Systems 65, 285–296 (1994)
10. Keller, J.M., Tahani, H.: Backpropagation Neural Networks for Fuzzy Logic. Information Science 62, 205–221 (1992)
11. Chung, F.L., Duan, J.C.: On Multistage Fuzzy Neural Network Modeling. IEEE Trans. on Fuzzy Systems 8(2), 125–142 (2000)
12. Duan, J.C., Chung, F.L.: Cascaded Fuzzy Neural Network Model Based on Syllogistic Fuzzy Reasoning. IEEE Trans. on Fuzzy Systems 9(2), 293–306 (2001)
13. Ishibuchi, H., Nii, M.: Generating Fuzzy If-Then Rules from Trained Neural Networks: Linguistic Analysis of Neural Network. In: Proc. IEEE International Conference on Neural Networks, pp. 1133–1138 (1996)
14. Huang, S.H., Xing, H.: Extract Intelligible and Concise Fuzzy Rules from Neural Networks. Fuzzy Sets and Systems 132, 233–243 (2002)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
16. Sekar, B.D., Dong, M.C., Shi, J., Hu, X.Y.: Fused Hierarchical Neural Networks for Cardiovascular Disease Diagnosis. IEEE Sensors Journal 12(3), 644–650 (2012)
17. Cui, Y.L., Dong, M.C.: Treat Opaqueness of Neural Networks System in Diagnosing Cardiovascular Disease via Rule Extraction. In: Recent Advances in Applied Computer Science & Digital Services, pp. 21–26. WSEAS Press, Japan (2013)

$kDMI$: A Novel Method for Missing Values Imputation Using Two Levels of Horizontal Partitioning in a Data set

Md. Geaur Rahman and Md Zahidul Islam

Center for Research in Complex Systems, School of Computing and Mathematics
Charles Sturt University, Bathurst, NSW 2795, Australia
{grahman,zislam}@csu.edu.au

Abstract. Imputation of missing values is an important data mining task for improving the quality of data mining results. The imputation based on similar records is generally more accurate than the imputation based on all records of a data set. Therefore, in this paper we present a novel algorithm called kDMI that employs two levels of horizontal partitioning (based on a decision tree and k-NN algorithm) of a data set, in order to find the records that are very similar to the one with missing value/s. Additionally, it uses a novel approach to automatically find the value of k for each record. We evaluate the performance of kDMI over three high quality existing methods on two real data sets in terms of four evaluation criteria. Our initial experimental results, including 95% confidence interval analysis and statistical t-test analysis, indicate the superiority of kDMI over the existing methods.

Keywords: Data pre-processing, data cleansing, missing value imputation, EM algorithm, Decision Trees.

1 Introduction

Data sets often have missing/corrupt values in them due to various reasons including equipment malfunctioning, human errors, and faulty data transmission [5, 12]. If an organization does not take extreme care during data collection then approximately 5% or more missing/corrupt data may be introduced in the data sets [10, 14, 18]. The application of various data mining alorithms, such as the classification and clustering algorithms, on the data sets having missing/corrupt values is likely to produce inaccurate results which may make the data sets less useful for the data miners [7]. Therefore, the imputation of missing values is an important task for effective data mining. For imputing missing values a number of methods have been proposed [1, 4, 12, 15, 19].

We consider a data set D_F as a two dimensional table with N records (or rows) and M attributes (or columns). The attributes of D_F can be of two types namely numerical (e.g. "9", and "9.5") and categorical (e.g. "Australia", and "China"). Let $R = \{R_1, R_2, \ldots, R_N\}$ be the records of D_F and $A = \{A_1, A_2, \ldots, A_M\}$ be the attributes of D_F. Also let r_{ij} be the value of the jth attribute of the ith record. By $r_{ij} =?$ we mean the value of r_{ij} is missing.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 250–263, 2013.

For imputing the missing value/s of R_i, the existing methods generally use the similarities of R_i with the other records of D_F, and the correlations of the attributes of D_F. Based on the nature of imputation, the methods can be classified into two broad groups, namely the global imputation and the local imputation [4].

Global imputation includes the methods that use global correlation structure of the whole data set for imputing the missing values of a data set. Some existing methods such as the Expectation Maximization based Imputation (EMI) [8, 15] fall in this group. EMI uses the correlations between the attributes having missing values and the attributes having available values, based on the whole data set, for imputing numerical missing values [15]. However, the methods that fall in this group are suitable only for the data sets that have strong correlations of the attributes within the whole data sets [17].

Instead of using a whole data set, the methods that belong to the local imputation group use only the similar records of R_i for imputing the missing value/s in R_i. The methods such as k Nearest Neighbor based Imputation (kNNI) [2], Local Least Squares Imputation (LLSI) [9], Iterative LLSI (ILLSI) [3], Bi-cluster based Iterative LLSI(IBLLS) [4] and Decision tree based Imputation (DMI) [12] fall in this group. For example, kNNI [2] first finds a user-defined k-NN records of R_i by using the Euclidean distance measure. For imputing a numerical missing value $r_{ij} \in R_i$, the method utilizes the mean value of the jth attribute within the k most similar records R_i. If the jth 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 better on the data sets having strong local correlation structure. However, the method can be found expensive for a large data set since for each record R_i having missing value/s it finds k-NN records of R_i by searching the whole data set. Moreover, the identification of a suitable value for k can be a challenging task.

Another local imputation based method called DMI [12] first divides a data set into a number of mutually exclusive horizontal segments obtained by the leaves of a decision tree (DT) obtained from the data set. The records belonging to each leaf (i.e. horizontal segment) are expected to be similar to each other. The correlations among the attributes for the records within a leaf are generally higher than the correlations of the attributes of the whole data set. Therefore, DMI applies an EMI algorithm on the records within a leaf instead of all records of the whole data set.

Fig. 1 presents a decision tree obtained from the CMC data set, which is available from the UCI machine learning repository [6]. A decision tree considers an attribute as the class attribute, which it aims to classify through the extracted patterns or logic rules. In the figure the ovals represent the nodes and the rectangles represent the leaves. The path from the root node to a leaf represents a logic rule. Each leaf contains a set of records (i.e. a horizontal segment of the data set) that follow the logic rule of the leaf. Some leaves have all records with the same class value. These leaves are called the homogeneous leaves. However, some other leaves contain records with different class values, and they are called the heterogeneous leaves. The records having different class values are likely to be dissimilar to each other. The dissimilarity of two records cause them to have different class values. For example, if two patients have different diseases (i.e. class values) such as “Liver Cancer” and “Hey Fever” then it is very likely that their health data are also very different. The existence of heterogeneous leaves within a

DT therefore indicates the grouping of dissimilar records together in a leaf. This may cause an imputation accuracy drop for DMI since the main idea of DMI is to apply EMI within the groups of similar records.

We argue that the performance of DMI can be improved if it applies EMI on the similar records of a leaf instead of all records of the leaf. For example, there are 369 records in Leaf 4 of the tree presented in Figure 1. Out of the 369 records 90 have class value "N", 126 have "L" and 153 have "S". The existence of different class values for the records indicate some dissimilarity among them. The records having the same class value are perhaps more similar to each other than the records with different class values.

Therefore, we propose to find similar records within a leaf and then apply EMI on them. If R_i has a missing value then we first identify the leaf where R_i falls in, and then find the k number of records (within the leaf) that are similar to R_i. Finally, EMI can be applied on the k records to impute the missing value in R_i. We argue that the records within a leaf are generally similar to each other. Moreover, we extract a group of records that are even more similar to R_i by finding the k most similar records of R_i within the leaf where R_i falls in. Therefore, applying EMI within this similar group of records should produce better imputation accuracy.

We propose a novel technique called $kDMI$ which considers two levels of partitions. In the first level, kDMI divides a data set into a number of mutually exclusive horizontal segments that are obtained by the leaves of a decision tree built on the data set. In the second level, it finds the k number of records that are the most similar to R_i, among all records of the leaf where R_i falls in. Additionally, $kDMI$ automatically identifies the best value for k by using our novel algorithm (see Algorithm 1). The method finally imputes the missing values of R_i through applying the EMI algorithm on the best k-NN records.

In this paper we have the following novel contributions: 1) The consideration of two levels of horizontal partitioning in order to improve the imputation accuracy, 2) A novel algorithm to automatically find the best k-NN records, of a record R_i, within a leaf, 3) A set of experimentations indicating a higher imputation accuracy by the proposed technique than the existing methods, and 4) The execution time complexity analysis pointing a lower time complexity for $kDMI$ than the other existing local imputation methods. The existing methods, such as IBLLS [4], ILLSI [3] find k-NN records by searching a whole data set whereas kDMI finds k-NN records by searching a leaf resulting in a better time complexity for kDMI.

Based on four evaluation criteria namely co-efficient of determination (R^2), index of agreement (d_2), root mean squared error ($RMSE$) and mean absolute error (MAE) we evaluate the performance of kDMI over the performances of three high quality existing methods namely DMI, EMI, and IBLLS on two real data sets namely CMC and Autompg. In the experiment we use 32 missing combinations for simulating missing values. The initial experimental results, including 95% confidence interval analysis and statistical t-test analysis, indicate a clear superiority of kDMI over the existing methods.

The organization of the paper is as follows. Our novel method kDMI is presented in Section 2. Section 3 presents experimental results and discussions, and Section 4 provides the concluding remarks.

2 A Novel Imputation Method–$kDMI$

We first discuss the basic concepts of the proposed technique called $kDMI$ before introducing the algorithm in details.

2.1 Basic Concept

The EMI algorithm generally performs better on a data set having higher correlations for the attributes of the data set. The attribute correlations for the records within a leaf are generally higher than the attribute correlations of the whole data set [12]. Therefore, DMI [12] first builds DTs from a data set by using a decision tree (DT) algorithm such as $C4.5$ [11] and then horizontally divides the data set by the leaves where the records belonging to a leaf are considered as a horizontal partition. The records in each partition/leaf are expected to be similar to each other. DMI then applies an EMI algorithm in each leaf for imputing missing values within the leaf.

The existence of different class values in a heterogeneous leaf, indicates low similarities between the records. The low similarity among the records can drop the quality of the DMI imputation. However, even in a heterogeneous leaf the records that have the same class value are supposed to be similar to each other. We now explain the concepts using a real data set called CMC which is publicly available in the UCI machine learning repository [6]. We build a decision tree (DT) from the CMC data set by using the $C4.5$ algorithm [11] as shown in Fig. 1. Note that the natural class attribute of CMC is "Contraceptive Method Used" which has three class values namely "N", "L", and "S".

Due to the heterogeneity in a leaf such as Leaf 4, the average similarities among the 90 records that share the same class value "N" is expected to be higher than the average similarities of all 369 records within the leaf. This is also reflected in our empirical test on the CMC data set (Fig. 2). We first calculate S_D, which is the average of the similarities of all records with all other records in the whole data set. This is presented in the first column of Fig. 2. We then calculate S_l, which is the average of the similarities of all records within a leaf L_l with all other records within the same leaf. The average of all such similarities for all leaves (S_L) is presented in the 2nd column of Fig. 2. Clearly S_L is greater than S_D. We then calculate $S_{L,N}$, which is the average of the similarities for all records having the class value N within the leaves. This is presented in the 3rd column. Similarly, $S_{L,L}$ and $S_{L,S}$ are also calculated. $S_{L,L}$ and $S_{L,S}$ are the averages of the similarities for all records having the class value L and S, respectively. It is clear from the figure that the similarities among the records having the same class values within the leaves are higher than the similarities among the records within the leaves.

Therefore, we argue that the performance of DMI can be improved by applying EMI on only the records that are similar to the record R_i within the same leaf where R_i falls in, instead of applying EMI on all records of the leaf. Therefore we propose a method called kDMI which imputes missing values by partitioning the data set horizontally in two levels; the First Level partitioning and the Second Level partitioning.

2.2 The First Level Partitioning

In this step, we divide a data set D_F into two sub data sets namely D_C and D_I where D_C contains records without missing values and D_I contains records with missing

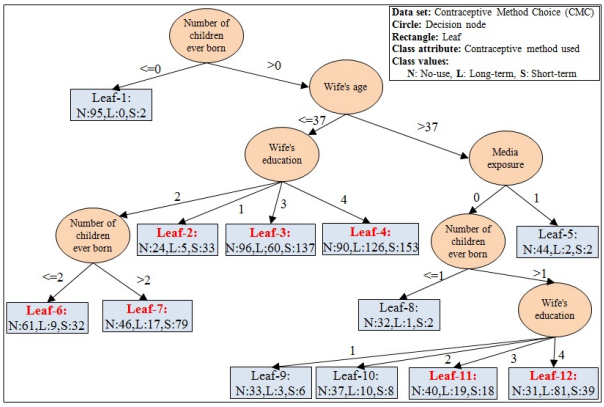


Fig. 1. A decision tree (DT) built from CMC data set

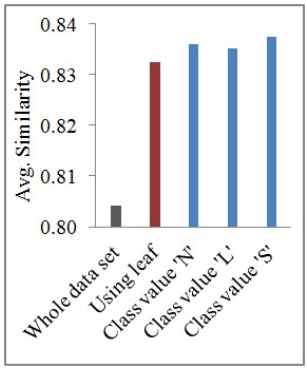


Fig. 2. Similarity analysis on CMC data set

values. For each attribute $A_j \in A$ having missing values (in D_I), we build a decision tree (DT) T_j through applying a decision tree algorithm such as $C4.5$ [11] on D_C by considering the attribute A_j as the class attribute. If A_j is numerical then we first generalize values of A_j in D_C into N_C categories where N_C is the square root of the difference between the maximum and minimum values of A_j. Note that for each T_j we have a set of logic rules $S_j = \{S_{j1}, S_{j2}, \ldots, S_{jp}\}$. For each logic rule S_{jl} of T_j, we generate a horizontal partition (i.e. a leaf) L_{jl} taking the records, from D_F, that satisfy the logic rule S_{jl}. If a logic rule states that $if(X = x_1)and(Y = y_1) \rightarrow (Z = z_1)$ then any record having $X = x_1$ and $Y = y_1$ is considered to satisfy the rule. Thus, we get mutually exclusive horizontal partitions of the data set where in each partition we have a set of records for the corresponding logic rule S_{jl}.

2.3 The Second Level Partitioning

For a missing value $r_{ij} \in R_i$, of a record R_i of D_I, we first identify the leaf L_{jl}, using T_j, where the record R_i belongs to. If R_i falls in multiple leaves (due to the missing value being tested for the leaves) then we use any of them. From the set of records belonging to L_{jl} we next find the best kNN records that are the most similar to R_i, by using our novel algorithm *BestKNN* (see Algorithm 1). Our *BestKNN* algorithm automatically finds a suitable value of k for the k-NN as follows.

The *BestKNN* algorithm takes as input the record R_i having missing values and the leaf L_{jl} where R_i falls in. The leaf L_{jl} has n number of records. In order to automatically find the best value for k, the algorithm first artificially creates a missing value (the actual value of which is available) in $r_{iz} \in R_i; z \neq j$. The algorithm then uses different k values (ranging from 2 to n) to impute r_{iz}. Finally, it takes the k value for which we get the best imputation accuracy. We next explain the process of finding the k value in more details as follows.

For each k value (ranging from 2 to n) the algorithm finds the set of kNN records (d_k) of R_i from L_{jl} by using the kNNI algorithm [2]. The algorithm then imputes the

missing value of r_{iz} by applying the EMI algorithm [15] on the k-NN records. Based on the imputed value and the actual value AV (which is known to us) the algorithm calculates the $RMSE_k$ [8]. Note that $RMSE_k$ values are calculated for all sets of k-NN records for the same r_{iz}. The best value for k is chosen from the set of kNN records d_k that produces the minimum $RMSE_k$ value. This horizontal segment of the best $k - NN$ records of a record R_i is considered as the second level partition.

Algorithm 1. Best$K$NN()

Input : A record R_i having missing value/s and a leaf Ljl having n records
Output : Best kNN records d_k of R_i from Ljl

$z = FindIndexOfAvailableValue(R_i)$; /* Find an index z of a numerical attribute for which the value of r_{iz} is available*/
$AV = r_{iz}$; /*Preserve the value $r_{iz} \in R_i$ into a variable actual value (AV)*/
r_{iz} =?; /*Artificially create missing values into r_{iz}*/
for $k = 2$ *to* n **do**
  $d_k \leftarrow FindKNNrecords(R_i, Ljl, k)$;/* Find the kNN records of R_i from Ljl by using the kNNI algorithm [2]*/
  $r_{iz} \leftarrow EMI(R_i, d_k)$;/* Impute r_{iz} by using the EMI algorithm [15] on the kNN records d_k*/
  $RMSE_k \leftarrow CalculateRMSE(AV, r_{iz})$; /*Calculate RMSE [8] between the actual value AV and the imputed value r_{iz}*/
end
Find a set of kNN records d_k as the best kNN records, for which the value of $RMSE_k$ is the minimum;
Return the best kNN records d_k;

2.4 Imputation

Once the best kNN records of R_i from the leaf L_{jl} are found, we impute the real missing value $r_{ij} \in R_i$ as follows. If r_{ij} is numerical then we impute r_{ij} by applying the EMI algorithm [15] on the best set of k-NN records. On the other hand, if r_{ij} is categorical we use the most frequent value of A_j from the best k-NN records as the imputed value.

By applying the similar approach, discussed in Section 2.3 and Section 2.4, we impute other missing value/s (if any) of R_i. Similarly we impute all other records having missing values in D_I.

Since kDMI finds the best kNN records from a leaf, we expect a lower computational complexity for kDMI than the existing k-NN methods such as IBLLS [4], ILLSI [3] where the existing methods find a suitable value for k by searching the whole data set. This is also reflected in the computational complexity analysis in the next section (see Table 3).

3 Experimental Results and Discussion

We implement our novel method kDMI and three high quality existing methods namely DMI [12], EMI [15], and IBLLS [4] for imputing missing values. We compare the performance of kDMI over the existing methods on two real data sets namely Autompg and Contraceptive Method Choice (CMC) that are publicly available in the UCI machine learning repository [6].

The Autompg data set has 398 records with a mixture of 5 numerical and 3 categorical attributes. The data set contains 6 records with natural missing values. For experimentation, we remove the records having missing values and thereby create a data set having 392 records without any missing values. Besides, the CMC data set has 1473 records with a mixture of 2 numerical and 8 categorical attributes. There are no records with natural missing values. Therefore, for CMC data set we use all 1473 records in the experiment.

We now artificially create missing values in the data sets by using four missing patterns namely Simple, Medium, Complex and Blended, four missing ratios: 1%, 3%, 5%, and 10%, and two missing models namely Overall and Uniformly Distributed (UD). The detailed description about the simulation of missing values is available in the literature [8, 12, 13].

Note that we have altogether 32 missing combinations (i.e. 4 missing ratios $\times$ 4 missing patterns $\times$ 2 missing models) for each data set. For each missing combination we create 10 data sets. Therefore, we create 320 (i.e. 32 missing combinations $\times$ 10 data sets for each combination) data sets for each data set.

We then impute the data sets by kDMI, DMI, EMI and IBLLS. For evaluating the performance of the methods we use four well known performance indicators namely co-efficient of determination (R^2), index of agreement (d_2), root mean squared error ($RMSE$) and mean absolute error (MAE). The detailed description about the evaluation criteria is available in the literature [8, 12, 16].

We now present the performance of kDMI, DMI, EMI and IBLLS on the Autompg and CMC data sets in terms of $RMSE$ and MAE in Table 1, where the bold values indicate that best results among the methods. In the table, we present the average values of evaluation criteria on 10 data sets for each missing combination. For Autompg data set, kDMI performs the best in 32 (out of 32) combinations in terms of both $RMSE$ and MAE. Moreover, for R^2 and d_2 (not shown in the table), kDMI performs the best in 30, and 28 combinations, respectively. DMI generally performs the second best. Similarly, for the CMC data set, kDMI performs the best in 28, 29, 27 and 31 combinations in terms of $RMSE$, MAE, R^2, and d_2, respectively. Like the Autompg data set, DMI performs the second best in the CMC data set.

We present the 95% confidence interval analysis on Autompg and CMC data sets for all 32 missing combinations in Fig. 3. Due to the space limitation we present the confidence interval analysis only for d_2. In Fig. 3 X-axis represents the missing combination IDs and Y-axis represents the d_2 values.

It is clear from the figure that kDMI has the better average values (and no overlap of confidence intervals) than DMI, EMI and IBLLS for most of the missing combinations except 6 combinations (marked by the circles) in the Autompg data set (Fig. 3(a)) and 2 combinations in the CMC data set (Fig. 3(b)). It is also clear from the figures that for the complex and blended missing patterns IBLLS in general performs poorly, whereas kDMI handles those missing patterns well. Even for high missing ratios kDMI performs almost equally well.

In Fig. 4, we present the aggregated performances based on R^2 for all methods in terms of missing ratios, missing models, and missing patterns on the Autompg (Fig. 4(a)) and CMC (Fig. 4(b)) data sets. The figures show that for all cases kDMI outperforms

Table 1. Performance of kDMI, DMI, EMI, and IBLLS on Autompg and CMC data sets in terms of $RMSE$ and MAE

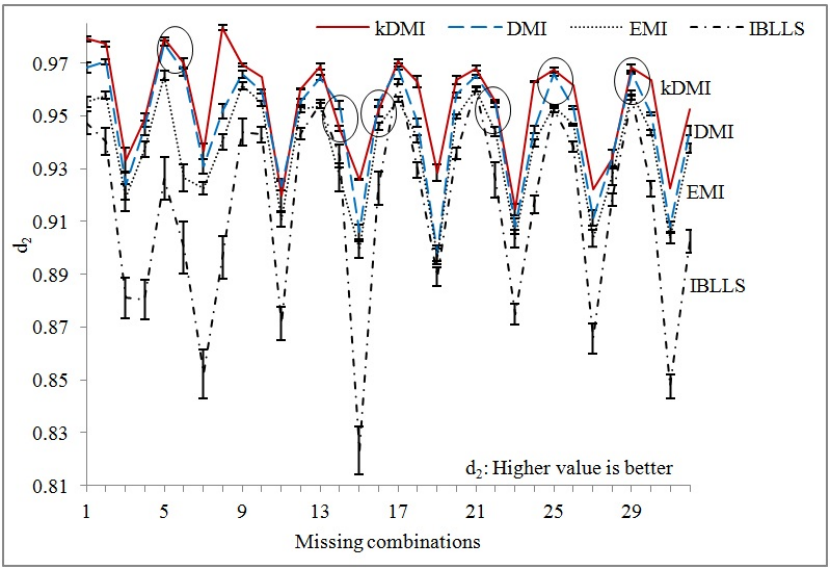
Missing combination			Id	Autompg data set								CMC data set							
				$RMSE$				MAE				$RMSE$				MAE			
				kDMI	DMI	EMI	IBLLS	kDMI	DMI	EMI	IBLLS	kDMI	DMI	EMI	IBLLS	kDMI	DMI	EMI	IBLLS
1%	Overall	Simple	1	**0.070**	0.078	0.090	0.087	**0.052**	0.059	0.069	0.068	**0.155**	0.160	0.168	0.187	**0.118**	0.119	0.128	0.136
		Medium	2	**0.065**	0.076	0.088	0.092	**0.053**	0.067	0.069	0.074	**0.160**	0.169	0.170	0.275	**0.118**	0.131	0.133	0.201
		Complex	3	**0.101**	0.101	0.104	0.137	**0.074**	0.075	0.079	0.101	**0.179**	0.187	0.188	0.349	**0.139**	0.142	0.145	0.271
		Blended	4	**0.088**	0.099	0.103	0.138	**0.067**	0.069	0.074	0.096	**0.174**	0.180	0.180	0.294	**0.137**	0.137	0.141	0.218
	UD	Simple	5	**0.071**	0.071	0.084	0.096	**0.053**	0.054	0.063	0.076	**0.146**	0.157	0.165	0.173	**0.111**	0.119	0.128	0.135
		Medium	6	**0.089**	0.090	0.119	0.133	**0.065**	0.067	0.093	0.100	**0.169**	0.177	0.184	0.285	**0.137**	0.140	0.143	0.214
		Complex	7	**0.093**	0.104	0.109	0.146	**0.074**	0.080	0.087	0.112	**0.166**	0.169	0.172	0.328	0.137	**0.136**	0.138	0.257
		Blended	8	**0.078**	0.084	0.091	0.113	**0.057**	0.062	0.068	0.091	**0.169**	0.178	0.183	0.284	**0.137**	0.142	0.147	0.214
3%	Overall	Simple	9	**0.082**	0.084	0.087	0.095	**0.059**	0.061	0.067	0.070	0.164	**0.160**	0.166	0.195	**0.119**	0.120	0.126	0.150
		Medium	10	**0.078**	0.091	0.094	0.106	**0.059**	0.066	0.071	0.072	**0.172**	0.174	0.177	0.268	**0.132**	0.133	0.137	0.198
		Complex	11	**0.112**	0.115	0.127	0.150	**0.085**	0.087	0.099	0.103	**0.200**	0.203	0.205	0.355	**0.153**	0.158	0.160	0.271
		Blended	12	**0.091**	0.095	0.097	0.107	**0.067**	0.070	0.073	0.072	**0.177**	0.180	0.186	0.290	**0.138**	0.138	0.142	0.214
	UD	Simple	13	**0.083**	0.088	0.097	0.098	**0.060**	0.063	0.072	0.070	**0.166**	0.166	0.170	0.203	**0.122**	0.126	0.131	0.159
		Medium	14	**0.087**	0.092	0.107	0.105	**0.063**	0.068	0.084	0.078	**0.180**	0.184	0.189	0.290	**0.135**	0.140	0.146	0.218
		Complex	15	**0.110**	0.128	0.130	0.166	**0.095**	0.099	0.101	0.113	0.191	**0.184**	0.185	0.352	0.154	**0.147**	0.148	0.270
		Blended	16	**0.093**	0.094	0.101	0.107	**0.069**	0.069	0.075	0.077	**0.174**	0.174	0.175	0.280	**0.132**	0.133	0.135	0.208
5%	Overall	Simple	17	**0.083**	0.084	0.096	0.090	**0.057**	0.061	0.072	0.065	**0.159**	0.161	0.166	0.190	**0.117**	0.123	0.128	0.145
		Medium	18	**0.097**	0.097	0.101	0.111	**0.068**	0.070	0.075	0.075	0.179	**0.178**	0.181	0.286	**0.130**	0.136	0.139	0.210
		Complex	19	**0.107**	0.129	0.130	0.141	**0.075**	0.099	0.100	0.103	**0.194**	0.198	0.199	0.357	**0.154**	0.155	0.155	0.270
		Blended	20	**0.086**	0.089	0.096	0.109	**0.061**	0.065	0.072	0.074	**0.173**	0.181	0.186	0.290	**0.135**	0.137	0.142	0.214
	UD	Simple	21	**0.083**	0.085	0.089	0.091	**0.059**	0.061	0.067	0.065	**0.168**	0.171	0.173	0.207	**0.124**	0.129	0.132	0.157
		Medium	22	**0.095**	0.095	0.101	0.107	**0.071**	0.071	0.077	0.080	**0.171**	0.173	0.178	0.271	**0.132**	0.133	0.138	0.200
		Complex	23	**0.113**	0.120	0.122	0.143	**0.091**	0.094	0.097	0.101	**0.180**	0.186	0.187	0.351	**0.142**	0.145	0.146	0.268
		Blended	24	**0.097**	0.102	0.105	0.123	**0.071**	0.075	0.079	0.085	**0.178**	0.180	0.183	0.280	**0.136**	0.137	0.140	0.206
10%	Overall	Simple	25	**0.083**	0.085	0.096	0.098	**0.057**	0.060	0.072	0.069	**0.166**	0.169	0.172	0.199	**0.125**	0.127	0.129	0.151
		Medium	26	**0.093**	0.097	0.102	0.111	**0.070**	0.072	0.077	0.076	**0.170**	0.179	0.182	0.279	**0.133**	0.138	0.140	0.206
		Complex	27	**0.112**	0.123	0.128	0.154	**0.090**	0.093	0.098	0.116	**0.182**	0.190	0.191	0.356	**0.145**	0.147	0.148	0.269
		Blended	28	**0.095**	0.108	0.113	0.123	**0.080**	0.081	0.087	0.084	**0.177**	0.180	0.182	0.282	**0.135**	0.137	0.139	0.206
	UD	Simple	29	**0.084**	0.086	0.093	0.095	**0.060**	0.063	0.070	0.068	**0.168**	0.170	0.174	0.198	**0.127**	0.129	0.132	0.151
		Medium	30	**0.095**	0.099	0.104	0.121	**0.071**	0.074	0.079	0.084	**0.178**	0.181	0.185	0.275	**0.135**	0.138	0.140	0.201
		Complex	31	**0.112**	0.123	0.125	0.161	**0.092**	0.096	0.099	0.112	0.193	**0.189**	0.190	0.341	0.152	**0.149**	0.150	0.260
		Blended	32	**0.094**	0.101	0.106	0.133	**0.072**	0.075	0.080	0.098	**0.176**	0.179	0.180	0.279	**0.131**	0.139	0.140	0.203
Score (Out of 32)				**32**	0	0	0	**32**	0	0	0	**28**	4	0	0	**29**	3	0	0

other methods in terms of R^2. Note that kDMI also outperforms other methods in terms of d_2, $RMSE$, and MAE (not presented here) for both data sets.

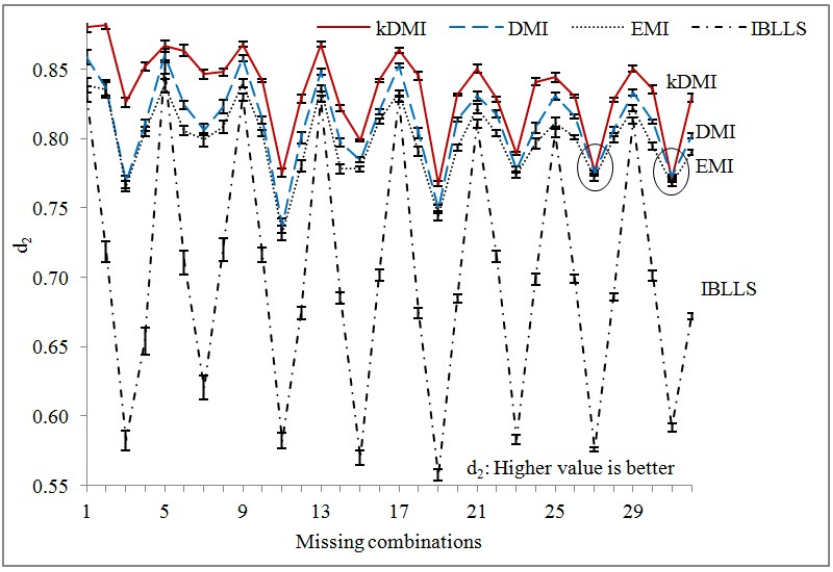
We also present the overall performances (i.e. the average accuracy calculated from the 320 data sets) based on R^2, d_2, $RMSE$, and MAE for the Autompg and the CMC data set in Table 2. For the data sets the overall imputation accuracy of kDMI is higher than the overall imputation accuracy of other methods. For the Autompg data set the overall imputation accuracy of kDMI in terms of R^2, d_2, $RMSE$, and MAE, are 0.834, 0.949,0.091, and 0.068, respectively. These results are better than the results of DMI, EMI and IBLLS. Similarly we get better imputation accuracy for kDMI on CMC in terms of R^2, d_2, $RMSE$, and MAE.

In Fig. 5, we present the percentage of the combinations (out of the total 64 combinations for the two data sets) where the methods perform the best. For example, kDMI performs the best in 95.31% combinations in terms of MAE (Figure 5(d)).

We also present a statistical significance analysis using t-test for all 32 missing combinations of all data sets in Fig. 6. The figure shows that kDMI performs significantly better than other methods at $p = 0.05$ in terms of all evaluation criteria for the Autompg and CMC data sets. The t-values are higher than the t(ref) values for most of the cases except those marked by the arrows.



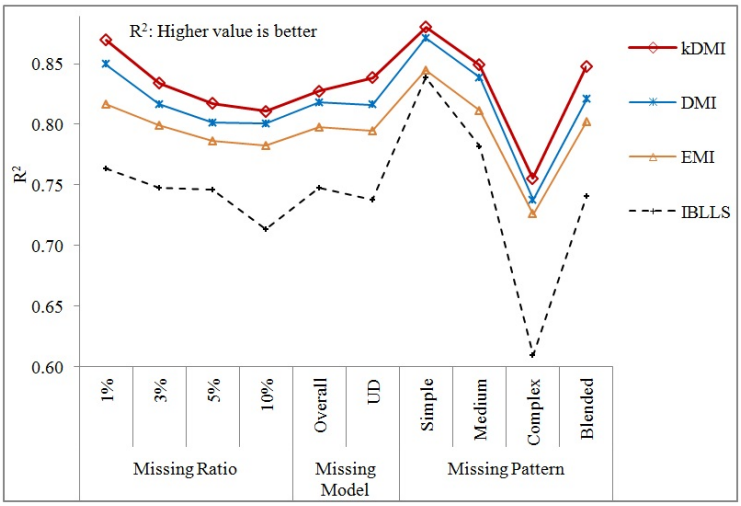
(a) Confidence interval on Autompg data set



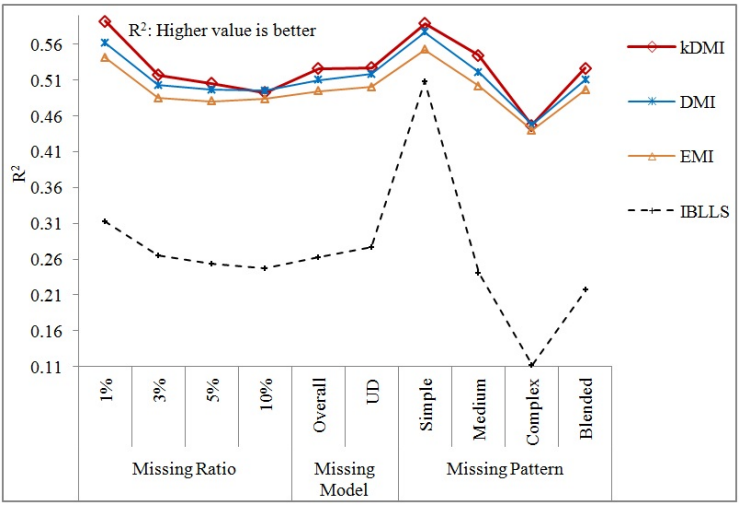
(b) Confidence interval on CMC data set

Fig. 3. 95% confidence interval analysis on Autompg and CMC data sets based on d_2

Among the four methods, only kDMI and DMI can impute both numerical and categorical missing values. Therefore, we now compare the performance kDMI with only DMI in terms of categorical imputation. Fig. 7 shows that kDMI performs better than DMI in terms of $RMSE$ and MAE for both Autompg and CMC data sets.



(a) R^2 on Autompg data set

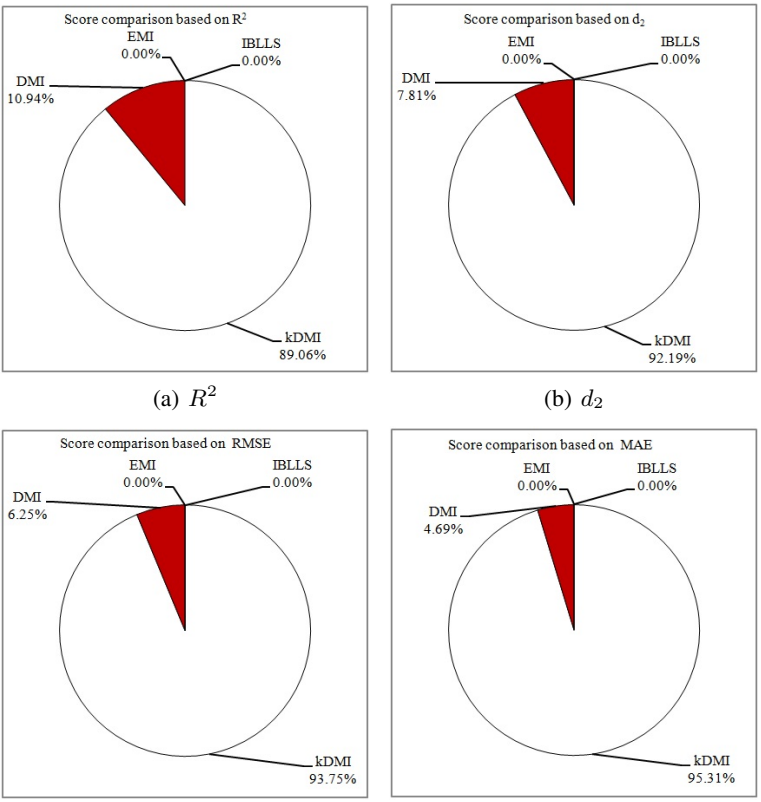


(b) R^2 on CMC data set

Fig. 4. Aggregated performance on Autompg and CMC data sets

Table 2. Overall average performance on Autompg and CMC data sets

Evaluation Criteria	Autompg				CMC			
	kDMI	DMI	EMI	IBLLS	kDMI	DMI	EMI	IBLLS
R^2	**0.834**	0.818	0.797	0.744	**0.527**	0.514	0.498	0.269
d_2	**0.949**	0.942	0.932	0.909	**0.834**	0.811	0.799	0.699
$RMSE$	**0.091**	0.097	0.104	0.118	**0.174**	0.177	0.180	0.277
MAE	**0.068**	0.073	0.080	0.085	**0.133**	0.136	0.140	0.208



(a) R^2 (b) d_2

(c) $RMSE$ (d) MAE

Fig. 5. Percentage of combinations for all data sets, where a method achieves the best result

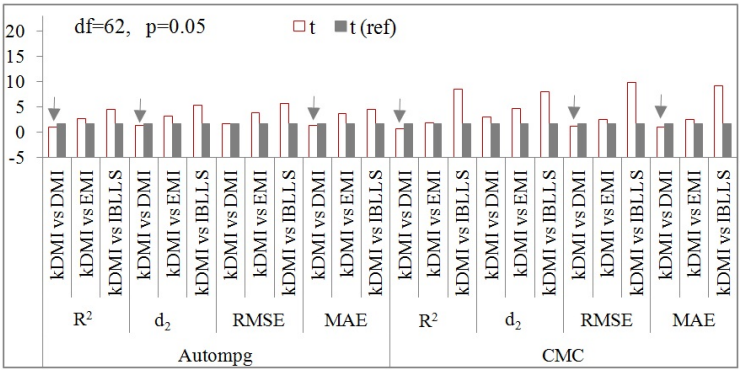


Fig. 6. t-test analysis on two data sets

Table 3 presents the average execution time (in milliseconds) for 320 data sets for the Autompg and CMC data sets. The experiments are carried out by using two different machines where Machine 1 is configured with 4 $\times$ 8 core Intel E7-8837 Xeon processors and 256 GB RAM, whereas Machine 2 is configured with Intel Core i5 processor having speed 2.67 GHz and 4 GB RAM. However, the experiments on a data set are done by using the same machine for all methods. Here, kDMI takes less time than IBLLS on both data sets, whereas it takes more time than EMI and DMI to pay the price of a significantly better quality of imputation.

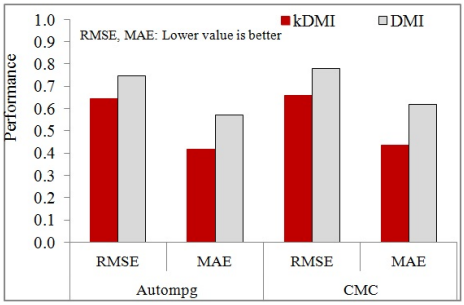


Fig. 7. Categorical imputation

Table 3. Average execution time (in milliseconds) of different methods on the two data sets

Data set	kDMI	DMI	EMI	IBLLS	Machine used
CMC	89,938	40,195	469	233,994	Machine 2
Autompg	6,826	2,215	18	8,861	Machine 1

4 Conclusion

For imputing the missing values of a data set we propose a novel method called kDMI which explores the best k nearest neighbor (k-NN) records of a record R_i having missing value/s. The kDMI technique uses two levels of horizontal partitioning in a data set. In the first level it divides a data set into a number of horizontal partitions that are obtained by the leaves of a decision tree. The records belonging to a leaf are generally similar to each other. However, a decision tree built from a real data set generally has heterogeneous leaves where a heterogeneous leaf contains the records with different class values (see Leaf 4 of Fig. 1). Records with different class values are likely to be different. Moreover, the average similarities of all records belonging to each class value within the leaves are higher than the similarities of all records of the whole data set. The former similarity is also higher than the similarity among all records belonging to each leaf (see Fig. 2).

Therefore, in order to achieving a higher imputation accuracy, we again partition a leaf horizontally (i.e. the second level partitioning) in a way so that the records belonging to the partition are the best k-NN records of a record R_i having the missing value/s.

A suitable value for k is determined automatically by our novel algorithm (see Algorithm 1). kDMI finally imputes the missing values by applying an EMI algorithm on the best kNN records.

The effectiveness of using the two levels of horizontal partitioning is also reflected in our initial experimental results. The experiments are carried out on two real data sets in order to compare the performance of kDMI with three high quality existing methods namely DMI, EMI, and IBLLS in terms of four evaluation criteria namely R^2, d_2, $RMSE$, and MAE. The kDMI technique outperforms other methods for all 32 missing combinations in terms of $RMSE$, and MAE on the Autompg data set (Table 1). According to the t-test in Fig. 6, kDMI performs significantly better than other methods at $p = 0.05$. Table 3 shows that kDMI requires less computational time than IBLLS since kDMI searches a leaf, instead of the whole data set, in order to find the best kNN records. Our future research plans include the extensive experiments of kDMI on more data sets, and further improvement of kDMI for reducing the time complexity.

References

1. Aydilek, I.B., Arslan, A.: A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm. Information Sciences 233, 25–35 (2013)
2. Batista, G., Monard, M.: An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence 17(5-6), 519–533 (2003)
3. Cai, Z., Heydari, M., Lin, G.: Iterated local least squares microarray missing value imputation. Journal of Bioinformatics and Computational Biology 4(5), 935–958 (2006)
4. Cheng, K., Law, N., Siu, W.: Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data. Pattern Recognition 45(4), 1281–1289 (2012)
5. Farhangfar, A., Kurgan, L., Dy, J.: Impact of imputation of missing values on classification error for discrete data. Pattern Recognition 41(12), 3692–3705 (2008)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml` (accessed July 7, 2013)
7. Han, J., Kamber, M.: Data mining: Concepts and techniques. The Morgan Kaufmann Series in data management systems (2000)
8. Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., Kolehmainen, M.: Methods for imputation of missing values in air quality data sets. Atmospheric Environment 38(18), 2895–2907 (2004)
9. Kim, H., Golub, G., Park, H.: Missing value estimation for dna microarray gene expression data: local least squares imputation. Bioinformatics 21(2), 187–198 (2005)
10. Maletic, J., Marcus, A.: Data cleansing: Beyond integrity analysis. In: Proceedings of the Conference on Information Quality, pp. 200–209. Citeseer (2000)
11. Quinlan, J.R.: Improved use of continuous attributes in C4. 5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
12. Rahman, M.G., Islam, M.Z.: A decision tree-based missing value imputation technique for data pre-processing. In: Australasian Data Mining Conference (AusDM 2011). CRPIT, vol. 121, pp. 41–50. ACS, Ballarat (2011)
13. Rahman, M.G., Islam, M.Z.: Data quality improvement by imputation of missing values. In: International Conference on Computer Science and Information Technology (CSIT 2013), Yogyakarta, Indonesia (2013)

14. Rahman, M.G., Islam, M.Z., Bossomaier, T., Gao, J.: Cairad: A co-appearance based analysis for incorrect records and attribute-values detection. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–10. IEEE, Brisbane (2012)
15. 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)
16. Willmott, C.: Some comments on the evaluation of model performance. Bulletin of the American Meteorological Society 63, 1309–1369 (1982)
17. Yan, D., Wang, J.: Biclustering of gene expression data based on related genes and conditions extraction. Pattern Recognition 46(4), 1170–1182 (2013)
18. Zhu, X., Wu, X., Yang, Y.: Error detection and impact-sensitive instance ranking in noisy datasets. In: Proceedings of the National Conference on Artificial Intelligence, pp. 378–384. AAAI Press; MIT Press, Menlo Park, CA; Cambridge, MA (2004)
19. Zhu, X., Zhang, S., Jin, Z., Zhang, Z., Xu, Z.: Missing value estimation for mixed-attribute data sets. IEEE Transactions on Knowledge and Data Engineering 23(1), 110–121 (2011)

Traffic Session Identification Based on Statistical Language Model

Xinyan Lou, Yang Liu, and Xiaohui Yu

School of Computer Science and Technology,
Shandong University, Jinan, Shandong, China
louxinyan880207@sina.com,
{yliu,xhyu}@sdu.edu.cn

Abstract. Session identification has attracted a lot of attention as it can play an important role in discovering useful patterns. A traffic session is a sequence of camera locations orderly passed by a vehicle to achieve a certain task. Based on the observations that both navigation regularity and temporal factor are crucial in determining the session boundaries, we propose an improved statistical language model which takes both factors into consideration in this paper. Extensive experiments are conducted on a real traffic dataset to testify the effectiveness of our proposal, and the result demonstrates its effectiveness compared to other alternative methods including the timeout method and the classic language model.

Keywords: session identification, timeout method, statistical language model.

1 Introduction

In recent years, the introduction of the high-definition camera technology makes capturing the vehicle information in real time possible. With the help of cameras installed on the main road, we can monitor and track the movement of vehicles. The information recorded by cameras may include licence plate numbers, timestamps, camera locations, etc. Figure 1 demonstrates an example of monitoring vehicles using such cameras. Obviously there may exist more than one cameras in one location, each of which is responsible for monitoring vehicle comes from one lane. In this way, once a vehicle passes a camera location, its movement can be observed, and such traffic data are later transmitted to the back-end data processing center continuously. By making the fullest use of these data, we can provide personalized Location Based Service. For example, [1] uses collaborative filtering to make targeted recommendations for users on locations where they might be interested to go and activities that they are likely to conduct. In addition, such information can benefit the Intelligent Transportation Systems constructed on traffic data by providing route-specific traffic information [2] and better advice to drivers [3], and even assist in making route predictions based on the traffic data and the given location[4].

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 264–275, 2013.

Fig. 1. The cameras for capturing vehicle information

The problem we studied in this paper is session identification in traffic data. A traffic session is a sequence of camera locations orderly passed by a vehicle to achieve a certain task. For example, during the journey from home to a supermarket, a vehicle passes three camera locations. We thus can consider the three locations formulate a session to achieve the task of shopping. Given a sequence of camera locations with time-stamps of a vehicle in one day, our objective is to identify which locations belong to one session, and further discover useful patterns and relations from rich data resource.

Although traffic session identification has broad application prospects, research in this field is rather limited. Compared to Web session identification and GPS route prediction which may share some similar characteristics to our work, traffic session identification has the following unique properties. First, the camera locations that a vehicle may pass is restricted, while a user can visit unlimited amount of web pages including both static and dynamic spots to get what he needs in web environment. Second, the sequences of camera locations that a vehicle passes at different times of the day might be similar or even the same, yet the users' browsing patterns have no such features. Third, due to the diversity of routes, the regularity of camera location sequences is local rather than global; therefore, a user's frequent sequences is not always frequent for the whole dataset. On the other hand, different from the portable GPS devices, the camera used to capture vehicle information is stationary. Besides, the sampling frequency of the camera is determined by the number of vehicles pass the camera location, while the GPS device's frequency is set by the user. Therefore, existing solutions to Web session identification and GPS route predictions cannot be applied directly to our problem.

In this paper, we propose an improved statistical language model for traffic session identification. This method combines the traditional language model with time influence factor, which not only considers the regularity of location sequences but also the time difference between neighbour locations. In addition, we compare our approach with alternative methods which only consider one influence factor to verify the effectiveness of the proposed method.

To the best of our knowledge, this is the first attempt to identify session in traffic applications. Our main contributions can be summarized as follows.

- We propose a novel traffic session identification problem using the data collected by cameras.
- We explain the principle of the traditional statistical language model and propose our time decay function.
- We combine the traditional language model with our time decay function to investigate the mutual effect of time factor and sequence regularity on traffic session identification.
- Extensive experiments are performed to verify the effectiveness of our proposed method.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related work. Section 3 introduces the datasets and the data preprocessing method. In section 4, a complete picture of the improved statistical language model is presented. Section 5 demonstrates the experimental results and discussion. The conclusion as well as future work is reported in section 6.

2 Related Work

There are several groups of research related to our work, namely, Web session identification, database session identification, and route prediction. Because Web session identification methods can also be applied to database session identification, so we just take Web session identification as an example. In this section, we present some main directions and major contributions of these works.

2.1 Web Session Identification

In web applications, a session is identified as a group of requests made by a user for a single navigation purpose [5]. The most commonly used method for Web session identification is called *timeout*, which identifies a session boundary between two requests when the time interval between the two requests is more than a predefined threshold. Catledge and Pitkow's work [6] claim that the mean time between each user interface event is 9.3 minutes. It assumes that most statistically significant events occurred within 1-1/2 standard deviations from the mean, so all events that occurred over 25.5 minutes apart are delineated as a new session. Instead of trying to find one particular session interval that would identify most session shifts, [7] calculates the probability of a particular time interval between two consecutive search activities containing a session shift. The optimal time intervals that can produce the lowest number of errors fall into the range of 8.6-15 mins in their tests. However, the task of determining the best threshold is still challenging as the value can be data dependent. In [5] and [8], Huang, Peng and An use statistical language model to solve the problem of session identification in Web log files. They consider each visited object as a basic unit, e.g., a word or character in natural language; therefore, the probability of object sequence can be estimated by the language model. Their experiments demonstrate that their approach is more effective than the timeout and other alternatives.

Similar to these works, our objective in this paper is to identify sequential camera locations that are related to a common task. However, due to the unique characteristics of traffic data as mentioned above, their methods cannot be applied directly to our work.

2.2 Route Prediction

Route prediction could be useful for giving the driver warnings about upcoming road situations, delivering anticipatory information and allowing the vehicle to automatically adapt to expected operating conditions [9]. In addition, it can be useful for providing better Location-Based Services, and improving social network and knowledge dissemination [4]. In [4], route patterns extracted from personal trajectory data with Continuous Route Pattern Mining (CRPM) are used to construct pattern tree for prediction. These patterns represent the trips that users frequently take, and each of them may contain more than one task-oriented sessions for each person.

Our work in this paper attempts to identify a sequence of camera locations passed by a vehicle to achieve a certain task. In a sense, a session is similar to the route pattern, which can reflect users' driving patterns at a finer granularity. Therefore, session identification in traffic data can be served as the basis for route prediction.

3 Observations

In this section, we will introduce our traffic dataset and test dataset, and give the formal definition of the traffic session identification problem.

The dataset we use is generated by traffic data center in a period of one month in Jinan, China. It contains 9939496 data records of 34947 vehicles collected from 164 camera locations. As a vehicle passes a camera location, the following information will typically be recorded and sent to the traffic data center: **license plate number**; **camera location number** which represents a specific crossing installed cameras; the **time-stamp** of the vehicle passes the camera location. The license plate number is similar to the IP address in Web log files, which may associates with several users. For the sake of simplicity, we assume that one plate number corresponds to one user in this paper. Therefore, users' traffic record can be represented as a set of ***vehicle-location-time*** triplets $<$ v, l, t $>$, each of which states the vehicle v passed the camera location l at time t. Given the vehicles' records gained from the data center, first we need to transform records into trajectories, each of which is the set of camera locations with time-stamp passed by a vehicle in one day. The trajectory records can also be expressed as the form of ***vehicle-date-sequence*** triples $<$ v, d, s $>$, which represents the vehicle v passes the camera location sequence s at date d.

Figure 2(a) shows an example of a user's records in two days. Figure 2(b) represents the trajectory form of records in Figure 2(a). As shown in Figure 2, the vehicle's daily trajectories show a high degree of temporal and spatial

regularity, and the trajectories of camera location passed by a vehicle in different time periods are usually similar or even the same. Nonetheless, this regularity dose not apply to all users.

Given a set of trajectories in the forms of vehicle-date-sequence triples < v, d, s >, in which the sequence s can be stated as $< l_1[t_1] - ... - l_n[t_n] >$, our objective is to identify which camera locations in s belong to one session. To obtain a golden-standard for the evaluation, we picked up 61456 records from the dataset collected by 164 camera locations and transformed them into 6036 trajectories. For each vehicle in those records, more than 70% camera locations it passed are attached with camera location name, latitude, and longitude. By making use of domain knowledge including both spatio and temporal information, our team finally identifies 27832 sessions manually.

license plate number	time stamp	camera location number
18587	2013/1/5 07:57:45	405
18587	2013/1/5 08:12:30	465
18587	2013/1/5 15:14:27	482
18587	2013/1/5 15:25:39	499
18587	2013/1/6 08:40:23	405
18587	2013/1/6 08:52:08	465
18587	2013/1/6 15:21:41	28
18587	2013/1/6 15:34:26	405

(a) An example of traffic data records

license plate number	date	trajectory
18587	2013/1/5	405[07:57:45]-465[08:12:30]-482[15:14:27]-499[15:25:39]
18587	2013/1/6	405[08:40:23]-465[08:52:08]-28[15:21:41]-405[15:34:26]

(b) The trajectory form of the left data records

Fig. 2. An example of data records and corresponding trajectories

4 Session Identification with Improved Statistical Language Model

In the task of traffic session identification, both driving regularity patterns and temporal features play important roles in determining the session boundary. To effectively capture such characteristics, in this section we first work on modeling each feature only, and then propose an integrated method to combine the two factors into a completed model.

4.1 N-Gram Statistical Language Model

N-gram language model has been sucessfully exploited in the task of Web session identification, and database session identification. Motivated by this idea, we adopt this strategy in our method for understanding the navigation regularities of drivers. In this section, we first give a brief introduction about the session detection method based on language model. This method uses the change of entropy in information theory to identify session boundaries dynamically by measuring the change of information in the sequence of camera locations.

When there are a series of locations visited consecutively for achieving a task, and such navigation pattern is observed constantly, the entropy of the corresponding location sequence tends to be low. However, when a new location is introduced afterwards, such change may lead to a dramatic increase in the entropy of the sequence because it is rarely visited before, and the new addition is not consistent with the navigation behaviour in the past. If the change in entropy passes a threshold, a session boundary could be placed before the new addition.

The original motivation for statistical language model is to solve the problem of speech recognition, where the goal is to know whether a word sequence is an understandable and meaningful sentence. In our traffic data, camera locations are passed sequentially in a particular order, similar to the word sequences that occur in natural language. If we consider each location as a basic unit, which may correspond to a word or character in natural language, we can then estimate the probability of camera location sequences using the language model. Given a sequence of camera locations $s = b_1, b_2, ..., b_M$, the probability of its occurence can be represented as:

$$\begin{aligned} P(s) &= P(b_1)P(b_2|b_1), ..., P(b_M|b_1, ..., b_{M-1}) \\ &= \prod_{i=1}^{M} P(b_i|b_1, ..., b_{i-1}), \end{aligned} \tag{1}$$

where M is the length of s. In this paper, we adopt the $n-1$ order markov assumption, which states that the probability of a location's appearance only depends on its at most $n-1$ preceding locations. Therefore, the probability of observing s can be estimated as:

$$P(s) = \prod_{i=1}^{M} P(b_i|b_{i-n+1}, ..., b_{i-1}), \tag{2}$$

where the subscript of b_w in $b_{i-n+1}, ..., b_{i-1}$ should always be larger than 0. The language model with $n-1$ order markov assumption is called *n-gram language model*. In this paper, we only consider the condition of $n = 1$. One most important issue in language model is how to estimate the conditional probability $\mathrm{P}(b_i|b_{i-n+1}, ..., b_{i-1})$. In general, if the sequence appears frequently, we can use the *Maximum Likelihood* method to estimate the values. That is,

$$P(b_i|b_{i-n+1}, ..., b_{i-1}) = \frac{c(b_{i-n+1}, ..., b_i)}{c(b_{i-n+1}, ..., b_{i-1})}, \tag{3}$$

where $\mathrm{c}(\alpha)$ denotes the number of times the string α occurs in the corpus.

However, data sparseness is an inherent property of language model in practice. Even for a very large data collection, the maximum likelihood estimation method does not allow us to adequately estimate the probability of rare but nevertheless possible sequences [10]. So we utilize Katz's smoothing to cope with the sparse data problem [11] in this work.

Finally, the quality of a given statistical language model can be measured by its entropy on a given camera location sequence s [12], where the entropy of the model on s can be calculated as:

$$entropy(s) = -\frac{1}{M} log_2 P(s), \tag{4}$$

This method utilizes the change in entropy to identify session boundaries dynamically. The change in entropy (denoted by Echange) is measured by the relative change in entropy values, which is defined as:

$$Echange = \frac{entropy(s') - entropy(s)}{entropy(s)}, \tag{5}$$

where s′ is a sequence of camera locations and s′ contains s plus the next location following s.

4.2 Time Influence Function

Based on our observations, temporal information has a significant impact on the behaviour of people, and people's driving routes usually display a great degree of time-related regularity. In our traffic dataset, every record is attached with an unique time-stamp. Intuitively, the larger the time interval between two camera locations, the smaller the probability that they belong to the same session. We thus propose a ***time influence function*** to measure the impact of time factor on session identification.

Given two camera locations b_i and b_j, whose time stamps are t_i and t_j respectively. The time influence function (denoted by $f(t_{i,j})$) can be written as:

$$f(t_{i,j}) = \frac{1}{1 + e^{-|t_j - t_i| + \beta}}, \tag{6}$$

where β is a non-negative number indicating the steepest gradient of this function. Due to the nature of this function, the time influence function value of $f(t_{i,j})$ on both sides of β changes rapidly. The meaning of β is to investigate the effect that different time intervals on session identification. So β is similar to the threshold which decides session boundary in timeout method. The values of this function have to be between 0 and 1. As can be seen from the above formula, the function value is proportional to the time interval.

4.3 Improved Statistical Language Model

In real life, most people prefer to choose a route that they are familiar with. Assume that there exists a sequence of camera locations in a session, and they are frequently visited in order without a long stay in between. Based on Equation (5) and (6), the change in entropy as well as the time influence function of this sequence will be relatively low. However, when the vehicle passes a new location that is not relevant to the original session, the introduction of this new location will cause an increase in the entropy of the sequence. Also, when the vehicle has a

long stay between the new location and the previous sequence, the time influence function value will be large. Both the increase in the entropy and the rise of time function value can be served as the signal for session boundary identification. So we combine these two factors to measure their mutual effect on session detection. If the mutual influence value passes a threshold δ, a session boundary could be places before new location. In the following steps, we explain how to combine Echange (change in entropy) with $f(t_{j,j+1})$ (time influence function) to measure their mutual effect on session detection.

Given a sequence s = b_1, b_2, ... , b_{i-1}, b_i, ... , b_M,

- We can get the probability of each sub-sequence $s_j = b_1, b_2, \ldots, b_{j-1}, b_j$ (j is not larger than n and we assume that the beginning of all sub-sequences is b_1) occurs and its corresponding entropy.
- Then, we compute the relative change in entropy values, i.e., Echange.
- We calculate the time influence function value of $f(t_{j,j+1})$. It is important to note that the process of the above computation is progressive, so we only consider the latest time interval when a new camera location added to former sub-sequence.
- Because these two scores, namely, Echange and $f(t_{j,j+1})$ are measured by different methods, and have different value ranges, so we normalize Echange using *min-max normalization* before we combine them. Suppose that min(Echange) and max(Echange) are the minimum and maximum values of all Echange. Min-max normalization maps a new value Echange to nor_Echange in the range (0, 1) by computing:

$$nor\_Echange = \frac{Echange - min(Echange)}{max(Echange) - min(Echange)}, \tag{7}$$

- For each sub-sequence, we use linear interpolation to weight nor_Echange and $f(t_{j,j+1})$ to measure the mutual effect value on session detection:

$$Value = \lambda * nor\_Echange + (1 - \lambda) * f(t_{j,j+1}), \tag{8}$$

where λ is a tuning parameter and it is not larger than 1.

5 Experiments

In this section, we first introduce the performance evaluation metrics used in our experiments, and then present the experimental results of our method as well as another two baseline methods, namely timeout method and n-gram statistical language model. Finally, we compare and analyse the results.

5.1 Performance Measurement Metrics

In this paper, we first use the widely adopted $F{-}Measure$ to evaluate the performance of traffic session identification. That is,

$$F - Measure = \frac{2 * precision * recall}{precision + recall}, \tag{9}$$

where a higher F-Measure value indicates a better overall performance.

In addition, we apply the metric of *Levenshtein Distance*, which is a traditionally used to measure the difference between two sequences and has been adopted to evaluate the performance of route prediction system in [4]. Informally, the Levenshtein distance between two sequences is the minimum number of single-character edits (insertion, deletion, substitution) required to change one sequence into the other. Compared with F−Measure, Levenshtein Distance evaluates the performance with a relative loose standard.

5.2 Experimental Results and Discussion

In this section, we evaluate our improved statistical language model by computing F−Measure and Levenshtein Distance. In addition, we compare and analyse the performance of our approach with the timeout method as well as the traditional n-gram language model.

Results of the Timeout Method. In our traffic data, the time intervals between two adjacent camera locations may vary from a few minutes to several hours. Intuitively, the larger the time interval between neighbour locations, the smaller the probability that they belong to the same session, and vice versa. Given a user's trajectory (time-stamped location sequence) $b_1[t_1]$-$b_2[t_2]$-...-$b_i[t_i]$-...-$b_n[t_n]$ and a redefined threshold t_δ. Assume that the time interval Δt between b_i and b_{i+1} is equal to $t_{i+1} - t_i$. If Δt is smaller than t_δ, then b_i and b_{i+1} belong to one session, otherwise, they will be divided into two different sessions.

In our experiments, we conducted experiments with several thresholds including 5, 10, 15, ..., 60 minutes. The results of these thresholds in terms of F−Measure and Levenshtein Distance are shown in Figure 3. The best F-Measure obtained is 77.20% under time threshold of 40 minutes and the smallest Average Levenshtein Distance is achieved when time threshold is set to 40 minutes. It is obvious that the performance of this method largely depends on the setting of time threshold values.

Results of the n-Gram Language Model Based Method. We then conduct experiments with the classic language model using the same settings as those in the timeout method. In Figure 4, we change the value of β from 0.29 to 0.33 at the step of 0.005, and it is clear that the best F-Measure value of (25.05%) and the smallest Average Levenshtein Distance of (3.1) are achieved when β is 0.315. The result is not even comparable with the timeout method, as the regularity of camera location sequences is local rather than global. That is, a user's frequent sequences is not always frequent for the whole dataset.

Results of the Improved n-Gram Language Model In the improved n-gram language model, there are three user-chosen parameters that provide the flexibility to fine tune the model for optimal performance. They include β, the weight λ and mutual influence value threshold δ. We now investigate how the choice of these parameter values affects the performance.

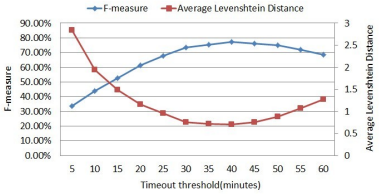


Fig. 3. The results of timeout method

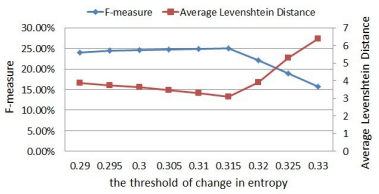
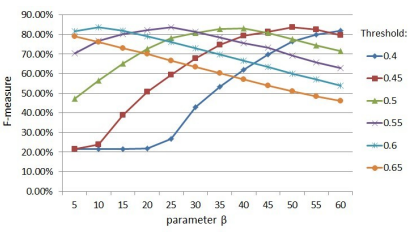


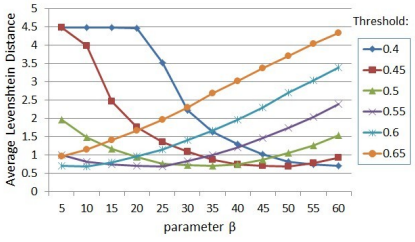
Fig. 4. The results of traditional n-gram language model

First, we utilize extensive experiments to study the effect on session identification of different weight λ. For each λ (0.1, 0.2, ... , 1), we have a try for every representative β (5, 10, ..., 60 minutes) and change the value of β from 0.4 to 0.65 at the step of 0.05 to get the optimal performance. We observe from figure 6 that the performance decreases as the λ increases and the model achieves its best performance when $\lambda = 0.1$. This suggests that the influence of time factor is bigger than the camera location regularity on session identification. When λ is equal to 1, this model degrades into the traditional language model, whose performance is poor because it completely without considering the time factor.

Then, we vary β and the mutual influence value threshold δ, with fixed λ ($\lambda = 0.1$) to study how these two parameters affect the performance. Refer to the performance of parameters in timeout method, we select 12 representative β values. As show in figure 5, the optimal performances of all β have no big difference. This suggests that the influence of different β values on the performance is stable as long as we find the corresponding optimal δ. However, for each β, the parameter δ that brings optimal performance is different. The larger the parameter β is, the smaller the δ we will need.



(a) The F-measure change with different β values



(b) The Average Levenshtein Distance change with different β values

Fig. 5. The results with different β values

5.3 Comparison and Analysis

Figure 7 illustrates the results of three different methods, where timeout(40) refers to the timeout method which threshold is 40 minutes, Trad. lang. model stands for the traditional language model with the threshold of entropy change is 0.315 and Improved 10 represents the improved language model which β is set to 10 minutes. This figure shows that our proposed method is significantly better than the traditional language model. This indicates that the time factor plays an important role in session identification. Although our method slightly outperforms the best timeout method, the result of the former method is relatively stable, in other words, it is robust to different (β).

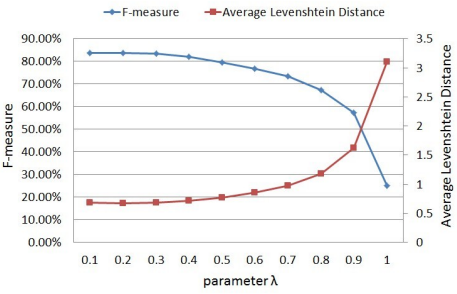


Fig. 6. The corresponding optimal performance with different λ

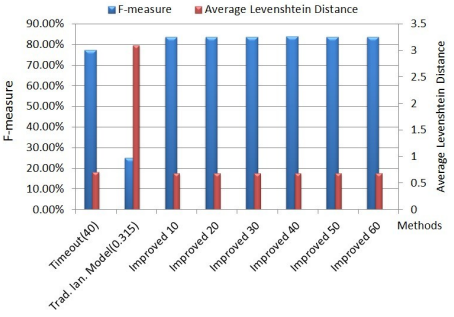


Fig. 7. Comparison of different methods

6 Conclusion

In this paper, we present an improved statistical language model to identify sessions in traffic data. This model not only considers the regularity of location sequences but also takes the time interval between neighbour locations into account. Experiments on test data show that our approach outperforms the traditional n-gram language model and timeout method in terms of F-Measure and Average Levenshtein Distance.

The results from our approach can be used to generate task-oriented route patterns, which are helpful for improving the precision of route prediction. Given the similarity between traffic data session identification with Web session detection, our work can be used to web applications.

This work will be extended in the following aspects. First, we will explore other factors that may affect the session identification, such as hot area and busy hour. Second, we will extract other auxiliary information, such as user similarity as well as camera location clusters to help improve the performance of session identification. Third, we can group the sessions into different classes, which can server as the frequent route patterns for route or next bayonet prediction.

References

1. Zheng, V., Zheng, Y., Xie, X., Yang, Q.: Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. Artificial Intelligence 184-185, 17–37 (2012)
2. Lee, W.H., Tseng, S.S., Shieh, W.Y.: Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system. Information Sciences 180(1), 62–70 (2010)
3. Torkkola, K., Zhang, K., Li, H., Zhang, H., Schreiner, C., Gardner, M.: Traffic Advisories Based on Route Prediction. In: Proceedings of Workshop on Mobile Interaction with the Real World, pp. 33–36 (2007)
4. Chen, L., Lv, M., Ye, Q., Chen, G., Woodward, J.: A personal route prediction system based on trajectory data mining. Information Sciences 181(7), 1264–1284 (2011)
5. Huang, X., Peng, F., An, A., Schuurmans, D.: Dynamic Web Log Session Identification With Statistical Language Models. Journal of the American Society for Information Science and Technology 55(14), 1290–1303 (2004)
6. Catledge, L., Pitkow, J.: Characterizing Browsing Strategies in the World-Wide Web. In: Proceedings of the 3rd International World Wide Web Conference, pp. 1065–1073 (1995)
7. He, D., Goker, A., Harper, D.J.: Combining evidence for automatic Web session identification. Information Processing and Management, 727–742 (2002)
8. Huang, X., Peng, F., An, A., Schuurmans, D., Cercone, N.J.: Session Boundary Detection for Association Rule Learning Using n-Gram Language Models. In: Xiang, Y., Chaib-draa, B. (eds.) Canadian AI 2003. LNCS (LNAI), vol. 2671, pp. 237–251. Springer, Heidelberg (2003)
9. Krumm, J.: A Markov Model for Driver Turn Prediction. In: Society of Automotive Engineers 2008 World Congress (2008) 2008-01-0195
10. Katz, S.: Estimation of Probability from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing 35(3), 400–401 (1987)
11. Wang, D., Cui, R.: Data Smoothing Technology Summary. Computer Knowledge and Technology 5(17), 4507–4509 (2009)
12. Bahl, L., Jelinek, F., Mercer, R.: A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence 5(2), 179–190 (1983)

Role Identification Based on the Information Dependency Complexity

Weidong Zhao, Haitao Liu, and Xi Liu

Software School, Fudan University, Shanghai

Abstract. Process mining mainly focuses on the control flow perspective at present. In comparison, role-based process mining stresses the importance of roles in business processes and their interactive relationships. Though some scholars come to pay attention to role identification, their studies are not sufficient in the analysis of role complexity. In this paper, a role coupling complexity metric based on information flow in the process is proposed, and the design structure matrix (DSM) is used for role identification in business processes. Then, some typical process logs are mined by an improved particle swarm optimization method. As the coupling complexity between roles is increasingly reduced, our method can recognize roles with lower complexity. Finally, experiments are performed to verify the effectiveness of the method.

Keywords: role identification, information complexity, DSM matrix, particle swarm optimization.

1 Introduction

Most process modeling methods are based on activities, which focus on activities, dependency relationships among them [1]. Compared with activity-based models, role-oriented process models highlight the most active organizational units and their interaction. Scholars have paid attention to the identification of these units and their relationships. Since participants are fundamental components of an enterprise, some scholars proposed a method to identify the social network according to the interaction among them. The more they interact with each other, the closer they are in the social network [2, 3]. Role engineering identifies roles in business processes by their permissions, since participants with the same or similar privileges are more likely to undertake similar activities [4, 5]. Jurgen et al. used bottom-up cluster analysis of participants' permissions to identify their roles [6]. However, these methods mentioned above didn't take process complexity into consideration and therefore may produce complicated role models. How to generate a simpler role model and reduce coordination cost as well as error rate during process execution requires further discussion.

At the same time, the concept of role is also introduced to the area of process complexity for new expansion. Yourdon evaluated the relationships among roles by considering data flow in role activity diagrams (RADs) [7, 8]. Phalp et al. analyzed the

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 276–287, 2013.

complexity of interaction among roles by calculating the proportion of internal activities to interactive activities in RADs [9]. However, these researches only focus on the interaction while lacking in-depth analysis of information dependencies among activities. Meanwhile, the importance of information flow between activities is neglected, which leads to insufficient analysis of process complexity.

Although complexity-based simple role identification has captured more and more attention, some problems remain unsolved. First, role data is not completely utilized. Most researches seldom put enough emphasis on information dependency and neglect the fact that information changes bring about changes of subsequent processes. Second, previous studies focused on the correctness of role identification devoting little attention to role complexity. However, role complexity directly affects understanding and execution of business processes. Furthermore, current role identification algorithms, for example, based on genetic algorithm, run slowly and inefficiently when the business process has a high complexity [10, 11]. As process becomes more and more complex in reality, scholars seek to mine simpler role-oriented processes.

In this paper, a role complexity metric based on information flow is proposed combined with an improved particle swarm optimization method. By using this algorithm, roles can be identified more efficiently. Besides, fitness function is introduced to reduce the complexity of information dependency.

The remainder of the paper is organized as follows: section 2 introduces a measure of information variability and sensitiveness for roles. Through the measurement of information processing capability and information dependency, the ability that a role employs information and the reliance of a role upon information are described. Then, the DSM-based particle swarm optimization method is introduced in section 3. Moreover, an improved particle swarm optimization method is proposed, which further improves the efficiency of process mining algorithm in section 4. Finally, we conduct some comparative experiments to show the effectiveness of the method.

2 Complexity of Information Dependency

From the role perspective, business processes are generally role-oriented: roles interact with each other to execute a process and achieve business goals. The interaction between roles can be considered as the cooperation among them for the same business goal.

The output of an activity is used by another activity, forming the dependency between them. The dependencies between activities are the basis of process operation and reflect the relationship between them as well. Dependencies can be identified by information flow, which reflects the cooperation between process participants. Therefore, closely interdependent activities can be encapsulated into a role. The internal activities taken by a role have higher dependency while activities executed by different roles have lower value. In other words, a process has higher relationship cohesion and lower information coupling.

The dependency and coupling of a process show how an activity outputs will affect subsequent activities. Herein, we define dependency relevancy among activities with

variability and sensitivity. Let $A = \{a_i | 1 < i < n\}$ be the set of all activities appearing in process logs, where n is the number of different activities. Let $a_i \rightarrow a_j$ be the direct dependency from a_i to a_j, indicating that there only exists the relationship that a_j depends on a_i but not vice versa. $a_i \rightarrow a_j$ means that there is a state transition from a_i to a_j in the process.

In this paper, the dependency degree between activities is evaluated by defining information variability and sensitivity on the basis of identifying dependencies between activities in business processes.

2.1 Variability and Sensitivity

Given the input information of a_i, variability is defined as the degree that how much its output for the subsequent activity a_j will change. High variability means a high possibility that the output will change. The subsequent activity a_j will receive the changed information from a_i. Sensibility defines the degree that how much a_j's output will change when the input of a_i changes. High sensibility indicates that small changes in the input from the preceding activity a_i will lead to greater changes of the output of the subsequent activity a_j.

Let r_{ij} be the information relevancy between a_i and a_j. Inf_{ij} is the information intensity between $a_i and a_j$. $\forall a_i \in A$, the input set $RI^i = \{r_{ki} | \text{Inf}_{ki} > 0, k = 1,2,\dots,n,\ k \neq i\}$, with the dimension $\dim(RI^i)$, and the output set $RO^i = \{r_{il} | \text{Inf}_{il} > 0, l = 1,2,\dots,h,\ l \neq i\}$ with the dimension $\dim(RO^i)$. The ordered pair (r_{ki}, r_{il}) is called a_i-centered information relevancy pair, denoted as c_{kl}^i.

$\forall c_{kl}^i$, how much r_{il} will change when r_{ki} changes? This is called the dependency relevancy of the relevancy pair, i.e. the relevancy value denoted as d_{kl}^i. High relevancy means that small change of r_{ki} will cause great change of r_{il}; otherwise it is called low relevancy. $\forall a_j \in A$, the variability of information relevancy r_{ij} between a_i and a_j relative to the input set RI^i of a_i is

$$\text{Alt}(r_{ij}) = \frac{\sum_{r_{xi} \in RI^i} d_{xj}^i}{\dim(RI^i)} \tag{1}$$

where $r_{xi} \in RI^i,\ and\ (r_{xi}, r_{ij}) \rightarrow d_{xj}^i$.

The sensibility of the information relevancy r_{ij} passed from a_i to a_j relative to the output set RO^j of a_j is

$$\text{Sen}(r_{ij}) = \frac{\sum_{r_{jy} \in RO^j} d_{iy}^j}{\dim(RO^j)} \tag{2}$$

where $r_{jy} \in RO^j,\ and\ (r_{ij}, r_{jy}) \rightarrow d_{iy}^j$.

2.2 Relevancy Evaluation

When computing the variability and sensibility, the relevancy can be evaluated by domain experts. Herein, the attribute hierarchical model (AHM) - an unstructured decision-making method - is introduced to evaluate the relevancy [12].

The analytic hierarchy process (AHP) divides a complex multi-goal decision problem into several sub-goals, which are further divided into multiple hierarchies of several indicators. AHP can effectively convert qualitative or half-quantitative problems into quantitative ones. However, as the number of indicators increase, the complexity of judgment matrix makes computation cumbersome. However, AHM does not need to calculate the eigenvalue and eigenvector of the matrix and thus is more convenient than AHP when there are lots of indicators. Therefore, in this paper, AHM is utilized to construct variability and sensitivity matrixes and calculate the dependency relevancy so as to handle complex information relevancy relationships. Here are the detailed steps:

(1) Determine evaluation objects and scales

All the relevancy degrees d_{kl}^{i} of relevancy pairs in the set C are re-labeled to form the evaluation set, denoted as $\mathrm{D} = \{d_1, d_2, \ldots, d_m\}$ where $\mathrm{m} = \dim(\mathrm{C})$. The relative relevancy degree of each pair can be determined by theoretical analysis or experiments.

In theory, the wider the range of output information is, the higher the degree that output change will be when input information changes. Consequently, the relevancy degree is higher. On the contrary, a narrower range of output means a lower degree of relevancy pairs. Similarly, a wide range of input information means a low relevancy degree and a narrow range of input information means a high one. For example, compared with reviewing engineering cost, the activity, calculating engineering cost has a relatively high relevancy degree. This is because the output range of calculation is quite wide and subtle changes of input will cause greater changes of output, while there are only two possible output values of reviewing engineering cost – passed or not, no matter how input information changes.

The types of information processing corresponding to activities mainly include creation, modification and passing. Information creation means activities create some new information, thus the corresponding relevancy degree is the highest. Activities can utilize input information and make some modifications. The relevancy degree of information modification is lower than information creation. Some other information may simply be passed to subsequent activities directly, which has the lowest relevancy degree. Based on the types of information processing, we can set different relevancy degrees for different information.

Moreover, information processing ability also has an effect on the relevancy degree and processing time indicates the ability to some extent. The relative relevancy degree b_{ij} is computed using Formula (3), where TO_i and TO_j indicate the range of output information for d_i and d_j respectively in process logs, while TI_i and TI_j denote the range of input information respectively. ω_i and ω_j are the relevancy degree weight of d_i and d_j respectively and determined artificially. t_i and t_j are the corresponding information processing time of d_i and d_j, that is, the output time of information minus the input time.

$$b_{ij} = \frac{TO_i}{TO_j} \times \frac{TI_j}{TI_i} \times \frac{\omega_i}{\omega_j} \times \frac{t_i}{t_j} \tag{3}$$

(2) Construct the comparison judgment matrix $(b_{ij})_{m\times m}$ using AHM.

(3) Convert the comparison judgment matrix into the attribute judgment matrix using Formula (4):

$$d_{ij} = \begin{cases} \dfrac{b_{ij}}{b_{ij}+1}, i \neq j \\ 0, \quad i = j \end{cases} \tag{4}$$

(4) Compute the relevancy degree d_i of every relevancy pair:

$$d_i = \frac{1}{m(m-1)} \sum_{j=1}^{m} d_{ij} \tag{5}$$

(5) Compute the variability and sensibility using Formula (1) and (2):

$$\mathrm{Alt}_{ij} = \mathrm{Alt}(r_{ij}), Sen_{ij} = \mathrm{Sen}(r_{ij}) \tag{6}$$

(6) Finally, compute the information dependency degree using Formula (7):

$$D_{ij} = \sqrt{\mathrm{Alt}_{ij} \times Sen_{ij}} \tag{7}$$

3 DSM-Based Particle Swarm Optimization

The design structure matrix (DSM) is used to construct the model for analyzing and optimizing the design and implementation of products, typically in the field of engineering, in order to improve design quality [13]. It is mainly conducted in the early stage of product design to represent input-output relationships between design activities and highly-coupled activity groups. In this paper, we extend the application of DSM to role identification in business processes.

3.1 DSM Based on Information Dependency

The DSM matrix consists of information dependency degree as its element. For example, D_{ab} is the information dependency degree between the activities a and b. The upper triangular matrix represents the information dependency degree in the forward information flow while the lower triangular matrix represents that in the information feedback flow.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 0 & D_{ab} & D_{ac} & D_{ad} & D_{ae} & D_{af} \\ D_{ba} & 0 & D_{bc} & D_{bd} & D_{be} & D_{bf} \\ D_{ca} & D_{cb} & 0 & D_{cd} & D_{ce} & D_{cf} \\ D_{da} & D_{ab} & D_{dc} & 0 & D_{de} & D_{df} \\ D_{ea} & D_{eb} & D_{ec} & D_{ed} & 0 & D_{ef} \\ D_{fa} & D_{fb} & D_{fc} & D_{fd} & D_{fe} & 0 \end{pmatrix} \end{array}$$

DSM performs well in hierarchical decomposition and modeling; it can decompose a larger structural matrix into smaller hierarchical one. Due to the characteristic of DSM, we can construct a DSM matrix according to the complexity of information coupling between activities and identify roles with higher inner cohesion and lower coupling in the matrix.

Three stages are needed to identify roles: DSM transformation, coupling activity set identification, and role identification.

(1) DSM transformation: transform the DSM matrix so that the feedback dependencies in the matrix are as close as possible to the diagonal. In this way, the feedback dependencies and their ranges can be reduced. Genetic algorithm and particle swarm optimization are two mostly-used algorithms for DSM optimization in recent years. Compared with genetic algorithm, particle swarm optimization can converge on the optimal solution faster and more efficiently. So particle swarm optimization is utilized to transform the DSM matrix in the paper.

(2) Identification of coupling activity sets: activities in the coupling activity set are closely connected. Since these activities depend on each other, the information dependencies among them are relatively high. Identification of coupling activity sets is the aggregation of strongly connected activities.

(3) Role identification: map the identified coupling activity set to each role.

3.2 Standard Particle Swarm Optimization

Particle swarm optimization (PSO) is a random search algorithm based on group coordination. It has been widely applied in many areas due to its strong robustness and good performance in parallel processing and global search [14, 15]. Particle swarm optimization is similar to genetic algorithm in many ways. They both initialize the population randomly, use the fitness to evaluate the solution quality and do some random searches. However, in PSO, particles have memory and share information with each other, making the whole particle swarm move to the best area smoothly. In PSO, only the global optimum P_g gives useful information to other particles, which forms a unidirectional information flow. In comparison, PSO converges to the optimal solution faster than genetic algorithm. In the case of role identification, if using PSO, all candidate results of role identification will approach the optimal result faster.

In the beginning, a group of random particles are initialized; then the optimal solution is obtained through iterations. During each iteration, particles get updated by

tracking two "bests". One "best" is the optimal solution found by the particles themselves, called the individual best solution P_i. The other is the optimal solution found by the whole population as yet, called the global best solution P_g. After finding the two solutions, particles update velocity and move to new locations. Besides, inertia weight factor is added to optimize the convergence speed of particles. The standard PSO algorithm is as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 \times rand(\,) \times (P_i^t - X_i^t) + c_2 \times rand(\,) \times \left(P_g^t - X_i^t\right) \quad (8)$$

$$X_i^{t+1} = V_i^{t+1} + X_i^t \quad (9)$$

where t is the generation of particles. For the t th and $t+1$ th generations of particle i, X_i^t and X_i^{t+1} are the corresponding locations respectively, indicating the role identification results. V_i^t and V_i^{t+1} are the velocity of the two generations, indicating the variation of identification results. P_i^t and P_g^t are the individual best and global best of the t-th generation respectively. The former best represents identification result of a specific particle with the highest fitness while the latter represents the optimal result of the swarm. Function *rand*( ) generates a random number in the range of (0,1). c_1 and c_2 are learning factors, both set to 2. ω is a none-negative value, called inertia weight, and usually is set to a function linearly increased with iteration times.

For standard PSO, moving direction of particles can be determined by three parts: original velocity V_i^t, which can make balance between the global and local optimal solutions; cognitive ability $P_i^t - X_i^t$, which means self-thinking of the particle i and contributes to the strong global search capability of i; and social knowledge $P_g^t - X_i^t$, which can lead i to the global optimal solution and share global information with each other. The importance of these parts can be determined by the coefficients ω, c_1 and c_2 respectively. For these parts, particles can converge to the optimal solution in the solution space.

In this paper, DSM matrixes are used as the input of PSO. Each DSM matrix, as an individual in the particle swarm, corresponds to the result of activity sorting. In the DSM matrix, D_{xy} represents the information dependency degree between the activities x and y. D_{xy}=0 means no information dependency between them. By reducing the number of none-zero elements below the diagonal of the matrix, we can divide the DSM into hierarchies. Therefore, the fitness function of PSO is defined as Formula (10):

$$\min F = \frac{1}{\sum_{x>y} D_{xy}} \quad (10)$$

4 Improving Particle Swarm Optimization

There are mainly two problems with standard PSO: particles start local search relatively slow and the performance will decrease as dimension increases. When the velocity of particles gets updated, the values of each dimension of the particles will get changed simultaneously. Among these changes, however, only a part of values will

approach the global best while others will move away from the global best, which decreases the performance [15]. To solve the two problems, new optimization of PSO is proposed.

Previous studies found that the decreasing strategy of the concave function performs better than a linear strategy, which performs better than the convex function. The decreasing strategy of the concave function enables particles to quickly begin local search with low velocity and thus improves the accuracy and convergence speed. Considering the inertia weight and decreasing concave functions, we set the inertia weight as Formula (11). ω decreases nonlinearly as iteration times increase. In this case, particle swarm will quickly begin local search at the early stage of iteration, ensuring the accuracy and speed of role identification.

$$\omega = 1/t \tag{11}$$

To solve the second problem, we can asynchronously process the velocities of particles in multiple dimensions when a certain condition is met. In other words, some dimensions of the particles are fixed and will not change any more, while for the other dimensions, the velocities will still get updated. Specifically, when *flag(t-1)*> μ and *random( )*> λ, set V_{id}^{t}=0. V_{id}^{t} is the velocity in the d th dimension of the t th generation for particle i. *flag(t-1)* is the times of continuous update of particle i's best fitness after the *t-1* th iteration. μ is a settable threshold; *random*( ) generates a random value in the range (0, 1]; λ is a linear function which decreases as iteration times increase:

$$\lambda = (1 - \frac{t}{T_{max}})/\,2 \tag{12}$$

where t is the number of iterations so far; T_{max} is the pre-determined maximum number of iterations. λ will decrease as iteration times increase, and accordingly the probability of the asynchronous update of velocities on multiple dimensions will dynamically change. This strategy can reduce negative influence of the multi-dimensional space on the algorithm's performance and largely improve particles' ability to find the optimal solution. For example, when the fitness of the role identification matrix M for a certain individual has been improved several times continuously, we can fix role identification results of some undertakers and change the others. This will vastly improve the role identification speed when there are many undertakers.

The global search ability of PSO is a little weaker than genetic algorithm. This is because when searching for the global best, particles may easily reach around the local best and get trapped in it. However, PSO converges to the optimal solution faster than genetic algorithm. To make full use of this advantage, we need to modify the search velocity of particles to improve the global search ability. In this paper, the mutation in genetic algorithm is introduced to optimize PSO. When a certain condition is met, the velocity of particles mutate: after iterations, the velocity in the update Formula (8) are assigned to a random value so that these particles will be able to jump out of the current local optimal solution.

When *flag(t-1)*> μ, the particle's velocity will mutate. μ is the threshold. When the times of continuous updates of particle *i*'s optimal fitness exceed the threshold, the particle may trap into a local optimum solution. By velocity mutation, the particle can jump out of local search.

Based on PSO, the process of role identification consists of the following steps:

(1) Initialize the particle swarm. Every particle in the particle swarm is an identification result while the velocity is the variation from a previous identification result to new one. Distribute the particles into solution space randomly, and assign an initial velocity to each of them. Calculate the current fitness P_i of *i* based on its location. The optimal fitness in the current swarm is denoted as P_g.

(2) Determine whether each particle' velocity needs a multi-dimensional asynchronous process or a random assignment by *flag* (*t*). As to the former, update the velocity with Formula (12). For the latter, assign a random value to the velocity. Otherwise, update the velocity by Formula (10) and recalculate its location.

(3) Calculate the fitness of all the particles and compare with the current P_i. If the fitness is larger, then update the particle's P_i and add one to *flag(t)*. Otherwise, set *flag(t*)=0.

(4) Compare with the current P_g of the swarm and update P_g if necessary.

(5) Check whether one of the termination conditions that the iteration times have reached the pre-determined T_{max} or the variance of the swarm's optimal fitness of consecutive *t* generations is smaller than the convergence standard is met. If the condition is not met, jump back to step (2). Otherwise, output the global optimal fitness and the global optimal solution.

The complexity of identifying the coupling activity set will largely decrease after sorting the DSM matrix by PSO. Detailed steps are as follows:

(1) For the re-sorted DSM matrix, if there exist non-zero elements in the lower triangular part of the matrix, i.e. $D_{xy} \neq 0 (x>y)$, the activities between *x* and *y* form the coupling activity set S_u. For each none-zero element, a coupling activity set is identified. In this way, we get the coupling activity sets $S_1, S_2, \ldots, S_n$.

(2) For the coupling activity sets identified above, if S_u and S_v have a common activity, merge them to generate a larger coupling activity set T_u. Repeat the process until there is no common activities between any two sets. $T_1, T_2, \ldots, T_n$ are the ultimate coupling activity sets.

(3) Map the identified coupling activity sets to the roles and summarize the characteristics of the roles.

5 Experiments

In the section, some experiments are conducted to verify the feasibility and performance of our method. We choose some event logs of 8 business processes from an insurance company for experimental analysis. Each process contains more than 20 instances, and we use the average fitness and running time for evaluation.

Figure 1 shows different fitness values for corresponding parameter μ. For process 1, 3 and 5, the fitness improves a lot when μ is set to smaller values as a result that larger value will decrease particles' ability of velocity mutation and make them trap into the local optimal solution. In other processes, different μ does not lead to a great

difference in the fitness because there are no or few traps of the local optimal solutions near the global optimal solution and it is easy to find the global optimal solution. When $\mu=1$, the fitness in all cases is quite low because frequent changes of the velocity make it hard for particles with the optimal fitness to gather around. Therefore, $\mu=1$ has a positive effect on searching for the global optimal solution.

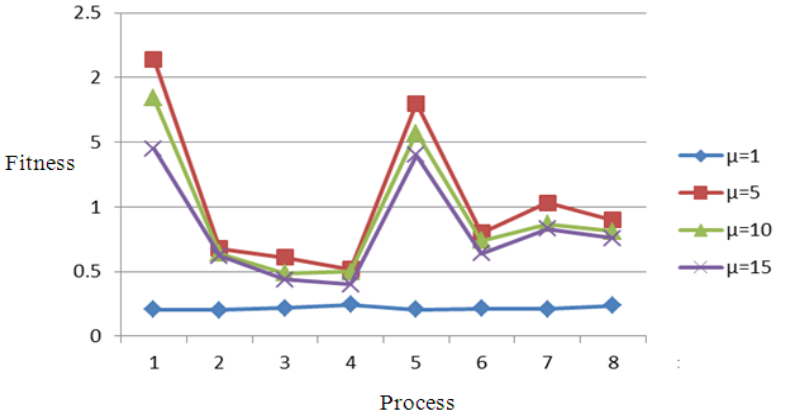


Fig. 1. Different μ and fitness

Figure 2 shows the fitness of three algorithms by analyzing the event log respectively: the algorithm proposed in this paper (Proposed algorithm), PSO(1-PSO algorithms, proposed by [16]) and genetic algorithm(2-genetic algorithm, proposed by [17]). For the processes 2, 4 and 7, our algorithm and PSO perform similarly while in the other processes, our algorithm performs better than PSO. Since standard PSO easily traps into the local optimal solution, we add a random disturbance factor into our algorithm to help particles jump out of the local optimal and find the global optimal solution. In all the cases, our algorithm is very close to genetic algorithm in the fitness because both algorithms are good at finding the global optimal solution.

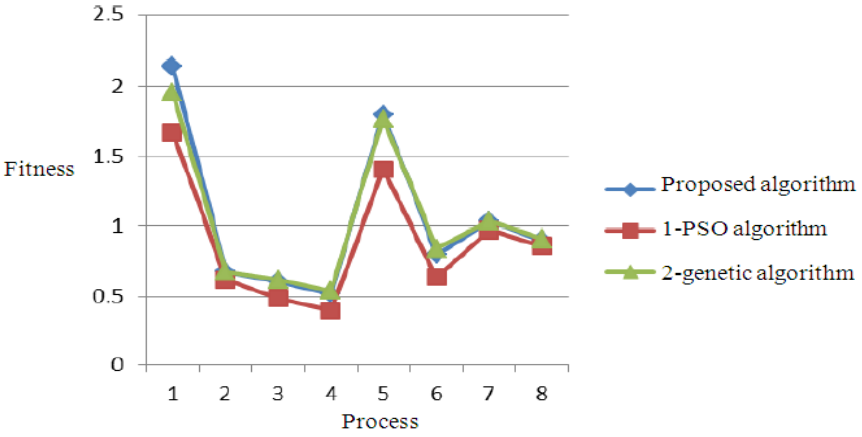


Fig. 2. Several algorithms and fitness

Figure 3 shows the temporal efficiency of our algorithm, PSO and genetic algorithm. As can be seen, for all the processes, PSO is more efficient than genetic algorithm. Because PSO will track the current optimal solution, our algorithm adopts the decreasing strategy of concave function so that the swarm will begin local search in the early stage of iteration. Therefore, our algorithm converges even faster than PSO.

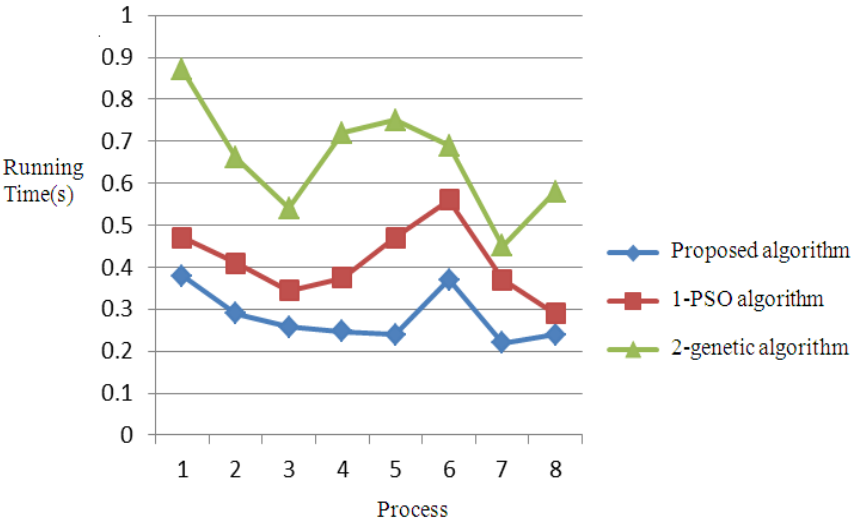


Fig. 3. Efficiency of several algorithms

6 Conclusions

In this paper, to identify simple roles with lower complexity, we propose an improved PSO algorithm to decrease role coupling. Compared with genetic algorithm, our algorithm converges faster and performs more efficiently. In addition, we introduce DSM to describe information dependency relationship between activities. By transforming the matrix, we can identify the coupling activity set, through which roles are identified. In PSO, nonlinear strategy helps solve the problem of local convergence to a certain degree and other improvement methods, like mixed strategies, will be considered in the future. By further improving PSO, the efficiency of role identification can be raised. Moreover, more work can be devoted to information communication among roles to identify more accurate and simpler roles.

Acknowledgments. The National Nature and Science Foundation of China under Grants no. 71071038 supports this work.

References

1. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology 7(3), 215–249 (1998)
2. Song, M., van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. Decision Support Systems 46(1), 300–317 (2008)
3. Ly, L.T., Rinderle, S., Dadam, P., Reichert, M.: Mining staff assignment rules from event-based data. In: Bussler, C.J., Haller, A., et al. (eds.) BPM 2005. LNCS, vol. 3812, pp. 177–190. Springer, Heidelberg (2006)
4. Colantonio, A., Di Pietro, R., Ocello, A., et al.: A formal framework to elicit roles with business meaning in RBAC systems. In: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, pp. 85–94. ACM (2009)
5. Frank, M., Streich, A.P., Basin, D.A., Buhmann, J.M.: A probabilistic approach to hybrid role mining. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 101–111. ACM (2009)
6. Schlegelmilch, J., Steffens, U.: Role mining with ORCA. In: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, pp. 168–176. ACM (2005)
7. Zhao, W., Dai, W., Wang, A., et al.: Role-activity diagrams modeling based on workflow mining. In: 2009 WRI World Congress on IEEE Computer Science and Information Engineering, pp. 301–305. IEEE (2009)
8. Yan, Z., Wang, T.: Role complexity analysis of business processes. Transactions of Beijing Institute Technology 28(3), 278–282 (2008)
9. Phalp, K., Shepperd, M.: Quantitative analysis of static models of processes. Journal of Systems and Software 52(2), 105–112 (2000)
10. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
11. Abdelsalam, H.M.E., Bao, H.P.: A simulation-based optimization framework for product development cycle time reduction. IEEE Transactions on Engineering Management 53(1), 69–85 (2006)
12. Qiansheng, C.: Attribute Hierarchical Model—A New Method of Unstructured Decision Making. Acta Scientiarum Naturalium Universitatis Pekinensis 34(1), 10–14 (1998)
13. Deng, X., Huet, G., Tan, S., et al.: Product decomposition using design structure matrix for intellectual property protection in supply chain outsourcing. Computers in Industry 63(6), 632–641 (2012)
14. Yan, X., Zhang, C., Luo, W., et al.: Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm. International Journal of Computer Science, 264–271 (2012)
15. Khokhar, B., Singh Parmar, K.P.: Particle swarm optimization for combined economic and emission dispatch problems. International Journal of Engineering Science and Technology 4(5), 2015–2021 (2012)
16. Das, S., Konar, A., Chakraborty, U.K.: Improving particle swarm optimization with differentially perturbed velocity. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 177–184. ACM (2005)
17. Dou, C., Lin, J.: Improved Particle Swarm Optimization Based on Genetic Algorithm. In: Wu, Y. (ed.) Software Engineering and Knowledge Engineering: Vol. 2. AISC, vol. 115, pp. 149–153. Springer, Heidelberg (2012)

Detecting Professional Spam Reviewers

Junlong Huang, Tieyun Qian*, Guoliang He, Ming Zhong, and Qingxi Peng

State Key Laboratory of Software Engineering, Wuhan University
Wuhan, Hubei, 430072, China
junlong-4444@163.com, {qty,glhe}@whu.edu.cn,
{mike.clark.whu,pengqingxi}@gmail.com

Abstract. Spam reviewers are becoming more professional. The common approach in spam reviewer detection is mainly based on the similarities among reviews or ratings on the same products. Applying this approach to professional spammer detection has some difficulties. First, some of the review systems start to set some limitations, e.g., duplicate submissions from a same id on one product are forbidden. Second, the professional spammers also greatly improve their writing skills. They are consciously trying to use diverse expressions in reviews. In this paper, we present a novel model for detecting professional spam reviewers, which combines posting frequency and text sentiment strength by analyzing the writing and behavior styles. Specifically, we first introduce an approach for counting posting frequency based on a sliding window. We then evaluate the sentiment strength by calculating the sentimental words in the text. Finally, we present a linear combination model. Experimental results on a real dataset from Dianping.com demonstrate the effectiveness of the proposed method.

Keywords: professional review spammer detection, text sentiment strength, posting frequency.

1 Introduction

With the rapid development of information technology, online shopping is becoming very popular. The customers tend to look at others' opinions before they make purchase decision. Since positive opinions often mean profits and negative opinions often mean losses, many businesses start to hire spammers to write fake reviews on different websites in order to promote their own products or discredit others' products. The fake reviewers and reviews are thus widespread across the web.

The opinion spamming is extremely harmful to the circumstance in e-commerce. More and more individuals and companies in e-business, as well as researchers, began to study how to detect and eliminate fake reviews/reviewers. Existing research has proposed methods to detect individual fake reviews [4-9,16,17,20], individual fake reviewers [14,18], and fake reviewer groups [15].

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 288–299, 2013.

The above existing works indeed help detect and prevent spam reviews to some extent. However, due to the financial incentives associated with fake reviews, the spammers never stop submitting spam opinions. Instead, they exert every effort to make their reviews more like true ones and their activities not deviating from normal behaviors. In other word, they are becoming more professional.

The professional spam reviewers have greatly improved their techniques of writing strategies in recent years. For example, they are consciously start using diverse expressions in their reviews. Below we give two instances extracted from a review site.

1. The restaurant is clean and the staff very helpful. Easy access to Metro.

2. Staffs were very friendly. The environments were clean. 5 mins walk to MTR.

We can see that the two reviews are almost the same but their expressions are quite different.This makes the similarity based detection method no more applicable. In addition, as some review sites may decline the multiple duplicate reviews to one specific product from the same id, the spam reviewers use agency or proxy to conceal the server. Overall, the spam reviewers and reviews are becoming more deceptive. It is a big challenge to build an accurate detection model to identify the professional reviewers and their reviews.

In this paper, we develop two new spam reviewer detection methods. One is based on reviewers' posting frequency and called frequency spammer detection model (FD). The other is based on the reviewers' emotional degree in their review texts and called sentimental spammer detection model (SD). FD is built by counting the number of reviews the reviewers posted in a specific sliding window. SD is built by counting emotional words (CSD).

We conduct experiments on a real dataset with a large number of reviewers and reviews extracted from Dianping (http://www.dianping.com/). We first select a small subset of samples from the various models for manual labeling by three evaluators. We then evaluate the accuracy of each evaluator by calculating the Cohen's Kappa coefficient and demonstrate the validity of each model by using the NDCG values of each model. The experimental results show that our proposed method can improve the accuracy of existing methods by a large margin. Finally, we learn a linear regression model from the manually labeling results and apply it on the entire data collection to detect review spammer.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 describes the basic notations and the main model for review spammer detection. Section 4 provides experimental results. Finally, Section 5 concludes the paper.

2 Related Work

Existing methods on opinion spam detection can be categorized into two main themes: to find spam reviews and find spam reviewers.

There are a number of works that study the detection of spam reviews. In [8], duplicate and near duplicate reviews were assumed to be fake reviews. Later, the top unexpected rules were mined to detect abnormal reviews and reviewers in [9]. Due to the lack of labeled data, Li et al. presented a two view semi-supervised method, co-training, to exploit the large amount of unlabeled data in [13]. Feng et al. examined different types of distributional footprints of deceptive reviews based on the assumption that a deceptive business entity that hired people to write fake reviews would necessarily distort its distribution of review scores [5]. Ott et al. developed three approaches to detecting deceptive opinion spam by integrating work from psychology and computational linguistics [7]. In [20], Yoo and Gretzel compared the language structure of deceptive and truthful hotel reviews and found that they differed in lexical complexity.

The detection of spammer was first attempted in [14]. The similarity between review texts and the deviations from the average rating were identified as characteristic behavior of spammers, and four methods were presented to model such behaviors. Although the four models targeted the same product or product groups, essentially they still relied on duplications. In [18], Wang et al. proposed a new concept of a review graph to capture the relationships among reviewers, reviews and stores that the reviewers had reviewed, and then the interactions between nodes in this graph were explored to identify suspicious reviewers. In [15], Mukherjee et al. derived several behavioral models and relation models to detect fake reviewer groups.

In general, existing studies for spammer detection used reviewers' behaviors, text similarity, linguistics features and rating patterns. None of them takes the sentiment strength in review texts into account. Moreover, the posting behaviors of professional spammers have not been carefully checked before.

3 A New Framework for Professional Spam Reviewer Detection

3.1 Notations

We first list the notations to be used in the subsequent sections.

- $U = \{u_i\}$: set of reviewers
- $S = \{s_j\}$: set of restaurants
- $E = \{e_k\}$: set of ratings
- $V = \{v_k\}$: set of reviews
- $V_{ij} = \{ v_k \,|u(v_k)= u_i$ and $s(v_k)= s_j \}$: set of reviews from user u_i to restaurant s_j
- $V_{i*}= \cup_j V_{ij}$: set of all reviews by user u_i

3.2 Professional Spam Reviewer Detection Based on Sentiment Strength

Sentiment Lexicon Generation

We build our sentiment lexicon mainly based on HowNet [3], an online knowledge base unveiling inter-conceptual relations and intra-attribute relations of concepts as

annotations in lexicons of the Chinese their equivalents. There are 836 positive sentiment words and 3730 positive review words, and 1254 negative sentiment words and 3116 negative review words in HowNet.

HowNet includes most of the sentiment words in general sense. As our data are online reviews for restaurants, some of the sentiment words specific to catering are not included in HowNet. In this paper, we construct a supplement sentiment lexicon based on the positive correlation between the explicit rating scores and the textual comments. The main procedure consists in the following three steps.

- We select from the database the reviews whose rating scores on Tastes, Service, Environment, Overall are all above 4.0, and labeling them as reviews with strong positive sentiments. This set is called PosS. It contains 31959 reviews in total. Similarly, we get 12460 reviews with strong negative sentiments whose rating scores are all lower than 2.0. This set is called NegS.
- We use the feature selection metric χ^2 to select words from PosS and NegS, and sort all words in descending order of their χ^2 values. The rationale behind this step is to remove non-informative features.
- Among the top 1000 words in PosS and NegS, we manually choose 840 and 350 positive and negative sentiment words. Here the number of positive words is much greater than the negatives ones. This is reasonable because in the review domain there are always fewer criticisms than praises.

缺斤短两，食物中毒，充数，反胃，徒有虚名，不足，走下坡路，超差，以假乱真，半生不熟，差强人意，便宜货，吹嘘，坑人，门可罗雀，破破烂烂，参差不齐，言过其实，难吃，不要脸，价格高，压抑，残渣，欠佳，离谱，奔溃，误导，苍蝇，每况愈下，格格不入

满分， 太香了， 价廉，高档次，上档次，值得一提，高性价，不虚此行，价廉物美，应有尽有，香气扑鼻，无可比拟，过瘾，美食，对得起，久负盛名，高品质，座无虚席，流连忘返，原汁原味，细致入微，丝丝入扣，高品质，鲜嫩嫩，香香，绝对值，与众不同，赞不绝口，怦然心动，超赞

Fig. 1. Top 30 sentiment words special for catering (Top: negative, Bottom: positive)

Figure 1 shows the top 30 sentiment words in catering. The English translations for the first five negative and positive words in Fig. 1 are “giving short weight, food poisoning, stopgap, with an undeserved reputation” and “full mark, smelling extremely well, low price, worth to mention, high price/quality ratio”, respectively. We can see that all these words are very meaningful and specific to restaurants.

By merging the above specific sentiment words with the general ones in HowNet, we get a positive set with 5026 sentiment words and a negative set with 4520 sentiment words. The duplicates are removed and only one copy is kept.

Computing Sentiment Strength

When computing sentiment strength for a review, we adopt a frequency based method. Suppose the count of positive and negative sentiment words in a review v_k is n_{pos} and n_{neg}, respectively, the sentiment strength of a review v_k is the combined sentiment word count and is defined as:

$$n_s(v_k) = | n_{pos} - n_{neg} | \tag{1}$$

A review v_k will be classified as a strong sentiment review if its sentiment strength $n_s(v_k)$ is greater than a user defined threshold $minsize^s$.

For each reviewer u_i, we can build a review set V_i for all his/her strong sentiment reviews. A reviewer u_i's sentiment strength is thus defined as the total number of sentiment word count in V_i. Formally, we have:

$$C_i(u_i) = \bigcup_{v_k} \{n_s(v_k) \mid v_k \in V_i * \wedge n_s(v_k) \geq minsize^s\} \tag{2}$$

In our experiment, $minsize^s$ is set to 3. We evaluate this threshold value by conducting the correlation analysis between the sentiment strength of the review content and the explicit rating scores. Normally, if a rating score is higher than 4 or lower than 2, the sentiment strength in this review is strong. Results show that the correlation is greater than 0.7. Hence the value is confidential for use.

CSD: A Professional Spam Reviewer Detection Model Based on the Sentiment Strength

For each reviewer u_i, let V_i be all his/her strong sentiment reviews and C_i be the sentiment word count in V_i, as shown in Eq. (2), the reviewer u_i's *spam score* function based on sentiment strength can be defined as:

$$c_{s,num}(u_i) = \frac{C_i(u_i)/|V_i(u_i)|}{\max_{u_j \in U} C_i(u_j)/|V_j(u_j)|} \tag{3}$$

Finally, the professional spam reviewer detection model based on sentiment strength is defined as:

$$CSD = \{u_i \mid u_i \in U_{crank}, i \leq m\} \tag{4}$$

where U_{crank} is the reviewer list sorted by their spam score $c_{s,sum}(u_i)$ based on the sentiment strength in descending order.

3.3 Professional Spam Reviewer Detection Based on Posting Frequency

In a review site, the reviewers with high grade are usually more trusted by other users. Some stores tend to recruit such kind of reviewers to post reviews to promote their own products and discredit others' products. Accordingly, there exist some people who are engaged in training useids (http://www.wkabc.com/detail/ 193536.html). These kinds of trained userids are then sold to professional spammers or keeping on posting spam reviews. In this section, we focus on finding this particular type of spam reviewers.

We first introduce some rules for training userids. All the rules are extracted from the largest web forum (bbs.tianya.cn) in China. For example, in bbs.tianya.cn/post-763-492879-1.shtml, a new trainer is asked to follow the rules listed below:

1. *You should register as a female user, and your age is between (25-30);*
2. *You are now living in Shanghai, and you will review the restaurant and shopping in Shanghai;*
3. *You should review about five or six stores everyday;*
4. *One userid can earn 20 points per day, and 10 days is a cycle.*

From the above rules, we can find that the main method for training userids is to frequently post reviews in a short period. Hence we propose our second professional spam reviewer detection model. This model is originated from the one used in [14], which is developed for finding the very high ratings on single product group. We adapt it based on the spammers' posting frequency.

We first define the reviewer u_i's review set within a span of time w as follows:

$$E_{i*}(w) = \{v_{ij} \in V_{i*} \mid s_j \in S \wedge t(v_{ij}) \in w\} \tag{5}$$

We set two types of time window, i.e., *day* and *month* based on the two following assumptions:

- Every day, a professional spam reviewer should post a specific number of reviews to maintain his/her grade;
- Every month, a professional spam reviewer should continuously post a specific number of reviews to train his/her userid.

Let C_i^{day} be the review set whose number of posts is larger than a minimum size threshold of $minsize^{day}$.

$$C_i^{day} = \bigcup\nolimits_w \{V_{i*}(w) \mid\mid V_{i*}(w) \mid \geq minsize^{day}\} \tag{6}$$

The u_i's spam score based on his/her daily posting frequency is defined by:

$$c_{f,day}(u_i) = \frac{C_i^{day}}{\max_{u_i \in U} C_i^{day}} \tag{7}$$

Similarly, the u_i's spam score based on his/her monthly posting frequency can be defined as:

$$C_i^{month} = \bigcup\nolimits_w \{V_{i*}(w) \mid\mid V_{i*}(w) \mid \geq minsize^{month}\} \tag{8}$$

and

$$c_{f,month}(u_i) = \frac{C_i^{month}}{\max_{u_i \in U} C_i^{month}} \tag{9}$$

In our experiment, we empirically set the time window w to be a day and a month and $minsize^{day} = 3$ and $minsize^{month} = 5$ respectively. One can notice that $minsize^{month}$ is a bit larger than $minsize^{day}$. The reason is that the data we extracted are only those from Shanghai. In a long time period, the professional spammers often review restaurants in a number of different cities so as to deceive the review system. Hence

the total number of reviews in one month in one city is not particularly large. We integrate the above two methods and define the u_i's posting frequency based spam score as follows:

$$c_f(u_i) = \frac{1}{2}(c_{f,day}(u_i) + c_{f,month}(u_i)) \tag{10}$$

FD: A Professional Spam Reviewer Detection Model Based on the Posting Frequency

Based on the spammers' posting frequency, we derive our second professional spam reviewer detection model:

$$FD = \{u_i \mid u_i \in U_{frank}, i \le m\} \tag{11}$$

where U_{frank} is the reviewer list sorted by their spam score based on the posting frequency value of $c_f(u_i)$ in descending order.

3.4 A Combined Model for Professional Spam Reviewer Detection

In previous section, we present two professional spam reviewer detection models. The frequency based model FD is a direct model by analyzing the reviewers' activities. The sentiment strength based model CSD is an indirect one, since it needs first computing the sentiment strength of a review, and then converts it into a reviewer's spam score. These two methods analyze the characteristics of reviewers from different views. To professional spam reviewers, they usually occupy both of the properties, i.e., the high posting frequency and the strong sentiment strength. Hence we propose a combined model.

The combined spam score of a user u_i is a linear combination of his/her spam scores of posting frequency and sentiment strength. It is defined as:

$$c_{fc}(u_i) = \alpha c_f(u_i) + (1-\alpha)c_{s,num}(u_i)), \tag{12}$$

where α is a parameter for balancing the impact of FD and CSD model.

Finally, we define the combined model FC for professional spam reviewer detection:

$$FC = \{u_i \mid u_i \in U_{fcrank}, i \le m\} \tag{13}$$

where U_{fcrank} is the reviewer list sorted by their combined spam score $c_{fc}(u_i)$in descending order.

4 Experimental Evaluation

4.1 Dataset

We collect data from the website of Dianping (http://www.dianping.com/). Similar to Yelp in US, Dianping is a professional third-party review sites with social networking features in China. The first categories in Dianping are cities like Beijing, Shanghai.

The second categories are listings for storefronts such as restaurants and shops. The data used in this paper are reviewers and their reviews on the restaurants in Shanghai.

We mainly use three types of information, i.e., restaurant information, reviewer information, and review information. The restaurant information includes storied, location, average costs, average star value, and average ratings. Each reviewer in Dianping can post one or more reviews. Reviewer information consists of userid, registering time, location, the last login time, community grade, review votes, etc. The review information includes posting time, reviewer, textual comments, numeric rating score, star value. We remove those records with null attributes. Table 1 shows the statistic for the data sets used in our experiment.

For the purpose of comparison with the baselines, we also extract information for chain restaurants. Among the 265 restaurants, there are 83 chain restaurants of 31 brands which have the same name but different ids and locations.

Table 1. Statistic information for datasets

Data set	*Before Preprocessing*	*After Preprocessing*
U: set of reviewers	206586	204954
S：set of restaurants	278	265
V：set of reviews	493982	489989
E：set of ratings	493982	489989

4.2 Baselines

We use four types of baselines proposed in Lim et al. [14].

- TC1 – This is a model used to detect the spam reviewers in chain restaurants. If a reviewer posts multiple similar reviews to one store (restaurant), he/she might be a spammer. Here the similarity is defined as the cosine similarity between the textual vectors of two reviews.
- TC2 – This is a model also used to detect the spam reviewers in chain restaurants. In a time window *w*, if a reviewer posts multiple reviews all having high or low rates, he/she might be a spammer. The time window *w* is set to *3*, and the threshold for the number of high and low ratings is set to *3* and *2* respectively. Here the value of high rating threshold is larger than that of low threshold because there are more promoting reviews than discrediting ones.
- GD – This is a model based on deviations from the average ratings on the same restaurant.
- ED – This is a model also based on deviations. Different from GD, this model assigns various weights to reviews according to their posting time.

4.3 Manually Labeling and Evaluation

Both the number of reviews and reviewers in our data are extremely large. It is very difficult, if not impossible, to manually labeling all the data. Assessment based on small sampling is mainly used in information retrieval, which uses several queries to evaluate the results from search engine. We adopt this method in our experiment.

We first construct two small sample sets for labeling. One is the reviewer set of most likely spammers, and the other is the set of most unlikely spammers. The detailed procedure for building the sample sets consists of three main steps.

1. Calculating the spam score of each reviewer using the FD and CSD models proposed in this paper and other four models (TC1, TC2, GD, ED) introduced in previous work [14] for spam reviewer detection.
2. Linearly combining the results from the above six models, and getting a combined spam score.
3. Sorting the reviewers in descending order of their combined spam scores, and then selecting 40 top ranked and 40 bottom ranked reviewers for user evaluation.

We recruit three graduate students to examine the selected reviewers. All of the students are familiar with the usage of *Dianping* website. They work independently on spammer identification. The selected reviewer set is randomly ordered before they are forwarded to the student evaluators. That is to say, the evaluators do not know the relationship between the reviewers' order and their combined spam scores. The evaluators then independently label every reviewer either as "spammer" or "non-spammer". Table 2 shows the number of spammers and non-spammers labeled by the evaluators.

Table 2. Evaluation Results

	Evaluator1	***Evaluator2***	***Evaluator3***
# Spammers			
Evaluator1	38	34	36
Evaluator2	-	41	38
Evaluator3	-	-	44
# Non-Spammers			
Evaluator1	42	35	34
Evaluator2	-	39	33
Evaluator3	-	-	36

In order to evaluate the three evaluators' consistency in their judgements, we compute the Cohen's kappa [1] values of the evaluator pairs. The results are listed below:

$K(1, 2) = 0.8252$,

$K(1, 3) = 0.7512$,

$K(2, 3) = 0.7744$

We find that $K(1,3)$ and $K(2,3)$ are between 0.6 to 0.8, indicating a substantial agreement. Meanwhile, $K(1,2)$ is between 0.8 to 1.0, indicating an almost perfect agreement [2,12]. Hence the judgements among evaluators are consistent and effective and thus can be used as the standard in following experiments.

4.4 Results

- **Evaluation metrics**

We use the NDCG (Normalized Discounted Cumulative Gain) value [11] to compare the results from the model and those from the evaluators. NDCG is a popular measure for spam reviewer detection [14, 15]. It is defined as follows:

$$NDCG = \frac{DCG}{DCG\ of\ ideal\ ranked\ list} \tag{14}$$

where DCG is defined as:

$$DCG = rel_i + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{15}$$

and DCG of ideal ranked list is the ordered list from three students. In Eq. (15), rel_i is the number of votes for reviewer ui ($rel_i \in [0, 3]$, $p = 80$).

- **Results for the proposed model**

We first investigate the effects of parameter α in Eq. (12). Figure 2 shows the results for various settings for α.

From Fig.2, it is clear that the larger value results in a better performance, showing that the combined model FC benefits more from the frequency based model FD than the sentiment strength based model CSD. We can see also that FC reaches the highest performance at α value of 0.9. Hence 0.9 is used as the default setting in all our experiments below.

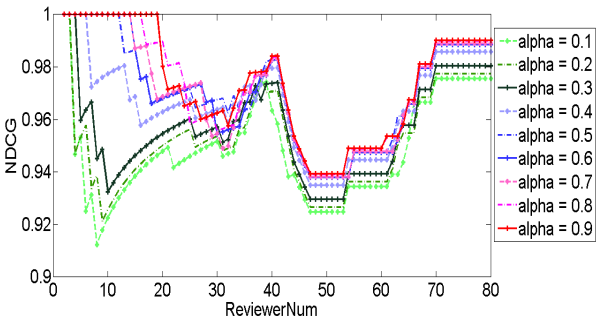


Fig. 2. The effects of parameter settings

Figure 3 shows the results for FD, CSD, and FC. Note that FD and CSD are actually the special cases of the combined model. If α is set to 0.0, then FC turns into CSD and if α is 1.0, FC turns into FD.

From Fig.3, we can see that FD performs better than CSD, and the combined model FC performs the best among three models. The reason can be due to that FC utilizes both the spammers' behavior patterns and the sentiment inclination of their writings. The combination of two kinds of models leads to a significant information gain and causes a large performance increase.

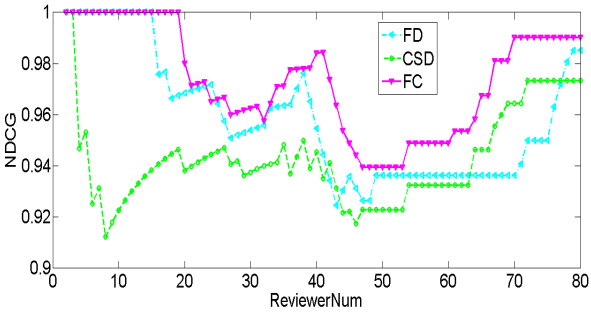


Fig. 3. The NDCG values of FD, CSD, and FC models

- **Comparison with the baselines**

Next, we compare our proposed model with the baseline methods. Figure 4 shows the results.

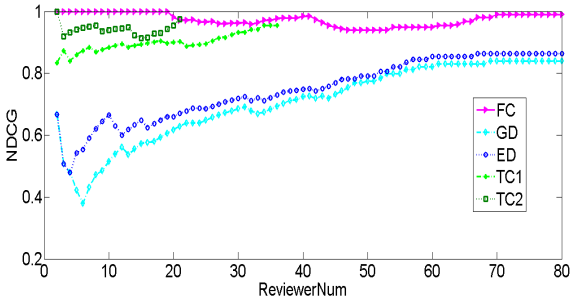


Fig. 4. Comparison with the baselines

It can be seen from Fig.4 that the proposed combined model FC generates the best ranking list. It performs better than any of the four baselines. The two detection models GD and ED, which are based on rating deviation, perform the worst. This indicates that the rating deviation models are impropriate for professional spammer detection. The TC1 and TC2 model are in the middle. Note that these two models are used to find the spammers in the chain restaurants, which are only a subset of the entire data. Hence their curves are shorter than those of other models.

5 Conclusion

In this paper, we develop two new models for detecting the professional spam reviewers. One is based on reviewers' posting frequency. And the other is based on the reviewers'

emotional degree in their review texts. We then linearly combine these two models and get an integrated one. We conduct experiments on a real dataset with a large number of reviewers and reviews extracted from Dianping. The experimental results show that our proposed model can improve the accuracy of existing methods by a large margin.

Acknowledgments. This research was supported in part by the NSFC Projects (61272275, 61272110, 61202036, U1135005), and the 111 Project (B07037).

References

1. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
2. Cohen, J.: Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. Psychological Bulletin 70(4), 213–220 (1968)
3. Dong, Z., Dong, Q.: http://www.keenage.com/html/c_index.html
4. Eason, G., Noble, B., Sneddon, I.N.: On certain integrals of Lipschitz-Hankel type involving products of Bessel functions. Phil. Trans. Roy. Soc. London A247, 529–551 (1955)
5. Feng, S., Xing, L., Gogar, A., Choi, Y.: Distributional Footprints of Deceptive Product Reviews. In: ICWSM (2012)
6. Gilbert, E., Karahalios, K.: Understanding Deja Reviewers. In: Proc. of ACM CSCW, pp. 225–228. ACM, New York (2010)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. of KDD, pp. 168–177 (2004)
8. Jindal, N., Liu, B.: Review spam detection. In: Proc. of WWW (Poster), pp. 1189–1190. ACM (2007)
9. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proc. of WSDM, pp. 219–230. ACM (2008)
10. Jindal, N., Liu, B., Lim, E.-P.: Finding Unusual Review Patterns Using Unexpected Rules. In: Proc. of CIKM (2010)
11. Järvelin, K., Kekäläinen, J.: IR evaluation methods for retrieving highly relevant documents. In: Proc. of SIGIR, pp. 41–48. ACM, New York (2000)
12. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics 33(1), 159–174 (1977)
13. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to Identify Review Spam. In: Proc. of IJCAI, pp. 2488–2493 (2011)
14. Lim, E.P., Nguyen, V.A., Jindal, N., et al.: Detecting Product Review Spammers Using Rating Behaviors. In: Proc. of the 19th CIKM, pp. 939–948. ACM, New York (2010)
15. Mukherjee, A., Liu, B., Glance, N.: Spotting Fake Reviewer Groups in Consumer Reviews. In: Proc. of WWW, pp. 191–200 (2012)
16. Ott, M., Cardie, C., Hancock, J.: Estimating the prevalence of deception in online review communities. In: Proc. of WWW (2012)
17. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In: Proc. of ACL, pp. 309–319 (2011)
18. Wang, G., Xie, S., Liu, B., Yu, P.S.: Review Graph based Online Store Review Spammer Detection. In: Proc. of ICDM (2011)
19. Xie, S., Wang, G., Lin, S., Yu, P.S.: Review spam detection via temporal pattern discovery. In: Proc. of KDD (2012)
20. Yoo, K.H., Gretzel, U.: Comparison of Deceptive and Truthful Travel Reviews. In: Information and Communication Technologies in Tourism, pp. 37–47 (2009)

Chinese Comparative Sentence Identification Based on the Combination of Rules and Statistics

Quanchao Liu, Heyan Huang, Chen Zhang, Zhenzhao Chen, and Jiajun Chen

Department of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
{liuquanchao,hhy63}@bit.edu.cn

Abstract. Opinions always carry important information of texts, but comparative sentence is a common way to express opinions. We describe how to recognize comparative sentences from Chinese text documents by combining rule-based methods and statistical methods as well as analyze the performance of these methods. The method firstly normalizes the corpus and Chinese word segmentation, and then gets the broad extraction results by using comparative words, sentence structure templates and dependency relation analysis. Finally we take CSR, comparative words and statistical feature words as classification features of SVM to accurately identify comparative sentences in the broad extraction results. The experiments with COAE 2013's test data show that our approach provides better performance than the baselines and most systems reported at CCIR 2013.

Keywords: Comparative Sentence, Chinese Comparative Sentence, CRF, CSR, SVM.

1 Introduction

Almost every day people are surrounded by all kinds of choices. In order to make better decisions, we usually compare items with the others which we are interested in. In big-data era today, we are easy to achieve a flood of useful information, comparing with traditional questionnaire approach. However, we are plagued by handling such a large amount of information, which would be time-consuming. Therefore, we need a comparative perspectives mining system to help us automatically obtain the comparison information between the two (or more things) from massive data.

The comparison is a universal feature of each natural language. It is not only an important aspect of grammar, but also an important method used for human thinking. We often compare two (or more) people or things when we make decisions. Therefore, we widely use multiple languages to express the comparison, such as English, Chinese and Japanese [1]. But the comparison approach embodied in the different languages is different. In the process of identifying English comparative sentence, [2] discusses how to recognize comparative sentence from English text, with support vector machine (SVM) and class sequential rule (CSR) algorithm to identify comparative sentences, it can achieve a precision of 79% and recall of 81%. On the basis of

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 300–310, 2013.

[2], [3] takes advantage of label sequential rule (LSR) algorithm to extract compare elements, and obtain satisfactory results. [4] and [5] obtain the relevant information by using Web search, and then compare two objects to get the relationship between them. Relying on the established rules, [6] presents a case study in extracting information about comparisons between running shoes and between cars, and use rule-based method to extract product mentions and comparisons from user discussion forums. Based on pattern recognition method, [7] proposes a weakly-supervised bootstrapping method based on indicative extraction pattern (IEP) to identify comparative questions, and they achieve F1-measure of 82.5% in comparative question identification.

However, Chinese comparative sentence refers to that the predicate contains comparative word, or the sentence has comparative construction. In the process of Chinese comparative sentence identification, [8] regards it as a binary classification problem on the basis of [2] and [3], and takes feature words, CSR as classification features of SVM. Based on the study of [8], extra CRF model is added in [9] to extract entity, the location and quantity of the entity is used as classification features of SVM to identify comparative sentences, and they achieve a precision of 96% and recall of 88%. What's more, [10] achieves Chinese comparative sentences through hierarchical network of concepts (HNC) theory.

Overall, the research on Chinese comparative sentence identification is still in its infancy. Because of the flexibility of Chinese, comparative sentence identification is more difficult. Currently, the methods solving the problem are mostly template-based matching and machine learning transforming the problem into classification problem. We design a three-step approach based on the combination of rules and statistics to identify Chinese comparative sentence in this paper:

(a) Design some rules for sentence structure to identify explicit comparative sentence.
(b) Compute the similarity of dependency relation in comparative sentence to obtain implicit comparative sentence.
(c) Extra features, containing comparative words, CSR and statistical feature words, are designed as classification features of SVM to identify comparative sentence.

The remainder of this paper is structured as follows. In Sec. 2, we briefly summarize Chinese comparative sentence. Sec. 3 describes our approach in detail to identify comparative sentence. Different experiments are done and experimental results are analysed in Sec. 4. Sec. 5 concludes our work..

2 Summary of Chinese Comparative Sentences

Generally speaking, comparative sentences are declarative sentences which contain compare and contrast, and require two or more objects on the semantic role. According to [11], modern Chinese comparative sentence is the sentence which contains comparative words or comparative constructions, and it consists of four comparative elements, namely comparative subject, comparative object, comparative points and

comparative results. Based on the approach of [9], comparative subject and comparative object are known as comparative entity while comparative points known as comparative attribute. Take "诺基亚N8的屏幕不如iphone的好" (Nokia N8's screen is not better than iphone's) for example, it is explicit comparative sentence, and is presented by quadruple form <诺基亚N8, iphone, 屏幕, 好> (Nokia N8, iphone, screen, better). In practical applications, above four elements sometimes don't appear simultaneously.

Due to its diversity and complexity, the definition and classification of Chinese comparative sentence has not been conclusive in academic world. In this paper, we use the criteria of COAE 2013 to classify comparative sentence.

(d) There are differences in ordinal relation between comparative entities, one is better than the other in comparative sentence, such as above instance.
(e) Just point out the differences but no good or bad.
"途安和毕加索的风格特点、细致程度以及技术含量都存在差异"
(There are differences in style, fineness and technology between Touran and Picasso)
(f) There have the same orientation or the similarity between comparative entities.
"诺基亚N8与iphone的通话质量差不多"
(Connection quality is approximately equal between Nokia N8 and iphone)
(g) There is only an entity which is the best or the worst among several entities in comparative sentence.
"iphone的屏幕是目前所有手机中屏幕最好的"
(Iphone's screen is the best of all)
(h) Compare two or more entities' features without comparative words, and don't point out which is better.
"诺基亚N8的屏幕材质是TFT的，但是iphone屏幕的材质是IPS的"
(The material of Nokia N8's screen is TFT while the material of iphone's screen is IPS)

In addition, some comparative sentences, which are controversial when human annotates or temporal-self comparison by time sequence, are not in the scope of our study. Such as sentence structure "the more…the more…", "…more and more…", "…less and less…" and so on. All in all, only when clearly master the definition, classification criterion as well as sentence structure, we would effectively and automatically identify Chinese comparative sentence.

3 Chinese Comparative Sentences Identification

The process of Chinese comparative sentence identification is shown in Fig. 1.

In order to solve the problem of non-standard data sets, we firstly normalize data sets, and then match sentence structure template and compute the similarity of dependency relation to separately identify explicit and implicit comparative sentences for the broad extraction. As a result, we obtain comparative sentence sets (A and B in

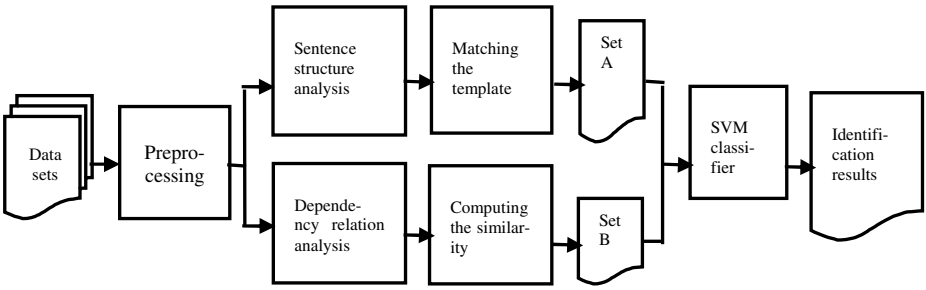


Fig. 1. The process of comparative sentence identification

Fig. 1) which has a low precision and a high recall. Finally, we take CSR, comparative words and statistical feature words as classification features of SVM to accurately identify comparative sentences in sets A and B. The identification results have a higher recall without losing the precision.

3.1 Data Preprocessing

Through analyzing data sets from COAE 2013, we conclude as follow:

(i) Taking sentence as unit in data sets, it has short text and sparse features.
(j) Existing imbalance between comparative sentence and non-comparative sentence, we need to process them into balanced corpus.
(k) Four comparative elements usually don't appear together in comparative sentences.
(l) High territoriality, extreme colloquialism.

Specific to the above characteristics, we design the process of data preprocessing as follows:

(m) All sentences are segmented and POS tagged using ICTCLAS2013[1], which contains the word segmentation of internet slang. And we use domain lexicon and comparative word dictionary to check the labeling results.
(n) Use Stanford Parser[2] to analyze sentence structure, and locate in comparative words, subject and predicate.
(o) Language Technology Platform (LTP)[3] is used to analyze dependency relation for data sets.
(p) Process the imbalanced corpus with entropy-based balance algorithm [12] and obtain balance corpus being close to one to one.

[1] http://www.nlpir.org/
[2] http://nlp.stanford.edu/software/lex-parser.shtml
[3] http://ir.hit.edu.cn/ltp/

3.2 Sentence Structure Templates Extraction

It is known that Chinese is a kind of highly flexible language. Most of Chinese language expressing comparative meaning contains comparative word, such as "比" (than), "亚于" (less than), "不如" (inferior to) and so on. [13] lists some common comparative words and comparative result words expressing the result which is better, as shown in Table 1.

Table 1. The words expressing comparative meaning in Chinese

Comparative words	Comparative result words
相比(compare with), 对比(contrast), 比较(compare), 不及(inferior to), 逊色于(inferior to), 劣于(inferior to), 弱于(weaker than), 等于(equal to), 等价于(equal to), 近似于(be similar to), 像(like), 犹如(as if), 如同(as), 优于(superior to), 好于(better than), 胜过(superior to), 超过(surpass), 区别于(different from)	差异(difference), 差别(difference), 区别(distinguish), 不同(diverse), 不一样(distinctive), 一样(the same), 媲美(match each other), 雷同(similar), 相同(the same), 不相上下(neck and neck), 旗鼓相当(neck and neck), 差(poor), 弱(weak), 欠佳(not good enough), 欠缺(defect), 不足(shortage), 劣势(inferior position), 好(good), 进步(progress), 领先(lead), 扩充(enlarge), 优势(superiority), 佼佼者(outstanding person), 出类拔萃(outstanding)

At the same time, some comparative sentences don't contain comparative word, we give the definition as follow:

Definition 3-1: *explicit comparative sentence*: contain comparative word and express the contrast between the two or more entities.

Definition 3-2: *implicit comparative sentence*: there is no comparative word in the sentence that is intended to compare two or more entities.

Through the analysis of data sets as well as the characteristics of comparative sentence, we summarize three kinds of sentence structure templates[4] with high coverage ratio for explicit comparative sentence.

(q) SS_1=… + VP (Keywords/Key Phrases) + …VA/ADJP…

It means that there is comparative word in the sentence, and the parent node of comparative word is VP, the other child nodes of VP are predicative adjectives (VA) or adjective phrases (ADJP).

(r) SS_2=… + VP (Keywords/Key Phrases) + …ADVP…

It means that there is comparative word in the sentence, and the parent node of comparative word is VP, the other child nodes of VP are adverb phrases (ADVP).

[4] Labeling with Stanford Parser.

(s) SS_3=… + NP (Keywords/Key Phrases) + …

It means that there is comparative word in the sentence, and the parent node of comparative word is NP.

3.3 Dependency Relation Similarity Computation

Explicit comparative sentence identification using sentence structure templates has a high recall. However, because of the complexity of implicit comparative sentence, we try to mine more information by analyzing dependency relation. The parser transforms Chinese sentence into structured dependency tree. The arcs in dependency tree reflect the relations between words in comparative sentence, and point out from core word to dependency word. The marks on the arcs represent dependency type [14]. Take implicit comparative sentence "诺基亚N8的屏幕材质是TFT的，但是iphone屏幕的材质是IPS的。"(The material of Nokia N8's screen is TFT while the material of iphone's screen is IPS) for example, its structured dependency tree generated with LTP is shown as follows:

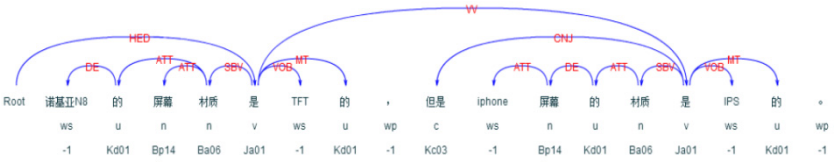


Fig. 2. The diagram of structured dependency tree generated with LTP

By analyzing lots of structured dependency trees like the above, we find that the two parts of implicit comparative sentence have similar syntactic structure and are combined with punctuation, turning words, coordinating conjunctions and so on. On the basis of [15], we give a fixed threshold to compute the similarity of dependency relations, and take the sentence, which similarity exceeds the threshold, as candidate of comparative sentence. Experimental results show that this method is very effective for compound sentence and transitional complex sentence.

3.4 Identifying Comparative Sentence with SVM

We describe our approach in Sec. 3.2 and Sec. 3.3 to identify comparative sentence. In our experiments, we find the approach has a low precision and high recall. In order to obtain the high precision and keep the high recall, we consider comparative sentence identification as a binary classification problem. [8], [9] and [12] adopt SVM classifier to identification comparative sentence and achieve good results. We propose three kinds of features for SVM classifier: CSR, comparative words and statistical feature words.

Class Sequential Rules

Because of the broad extraction containing explicit/implicit comparative sentence identification, our approach ignores the fine-grained level of word sequence. So we compensate for this defect by adding auxiliary features: CSR. Sentence structure has loose force of constraint and is suitable for the broad extraction, but CSR has strong force of constraint and is good at the accurate extraction. Based on the candidates of the broad extraction, we design CSR to take a further mining for improving the precision of the classification.

Sequential pattern mining (SPM) is one of the most important tasks in the field of data mining. CSR is suitable for SPM and its aim is to find the patterns which meet minimum support defined by user. [8] takes CSR as classification feature and proposes each clause as a sequence, as a result [8] achieve good performance for regular comparative sentence. But it would get half the results with double the effort for the colloquial/irregular comparative sentence. We choose different window size to solve this problem. At the same time, the sparse samples lead to that the values of the minimum frequency of CSR sequence as well as the median frequency sorting are 2. So the initial value is also 2. What's more, through the experiment we find that making each clause as the window size is not only inefficient but also lead to decreased accuracy. However, we can get a better effect when the window size is 5 in experiment.

It isn't good enough only to use CSR to distinguish comparative/non-comparative sentence. This is mainly because lexicon can not perfectly express the meaning of the sentence. [9] verifies that, so we need more semantic information to improve the classification features.

Comparative Words and Statistical Feature Words

Comparative words has been described in Sec. 3.2, we don't repeat them again. Statistical feature words are done on the balanced corpus. Through separately computing the distribution of a word w in intra-class or inter-class, we can get the distribution of w on the given data sets. We choose the word w with bigger intra-class information entropy and smaller inter-class information entropy as statistical features. Information entropy is defined as following:

$$H(w) = -\sum_{i=1}^{r} p(c_i|w) log_2 p(c_i|w) \quad (1)$$

Where $p(c_i|w)$ represents the probability that the document belongs to class c_i when word w appears in the document. When the value is greater, the word w will appear frequently in the class. It means the word w is more suitable to represent the class. Then we calculate information gain value for each word w, and set the threshold T to filter out the noise. Through three experiments, we find that $T = 12.6$ is better and finally obtain 3,000 statistical words.

4 Experiments

4.1 Performance Evaluation

Comparative sentence identification is evaluated by Precision, Recall, and F-measure. They are described as follows:

$$Precision = \frac{\#system\_correct}{\#system\_proposed} \tag{2}$$

$$Recall = \frac{\#system\_correct}{\#person\_correct} \tag{3}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Where $\#system\_correct$ is the correct result from system, $\#system\_proposed$ is the whole number of comparative sentences from system, $\#person\_correct$ is the number of comparative sentences that has been annotated correctly by people.

4.2 Data Set

For the current study of the comparative sentence is still very rare, and there is no more public valuable corpus. In order to carry out the experiment, on the basic of evaluation data from the COAE2013, we collect more which is derived from "*Zhongguancun Online*"[5] product review sites. The combined corpus contains three types of data: news, customer review and user forum, which are from the fields of cars and electronics. By manual annotation, we get the results as follow:

Table 2. Data Sets

Sentence type		Car field	Electronic field	In all
Comparative sentence	explicit	907	983	1890
	implicit	93	17	110
Non-comparative sentence		1000	1000	2000

4.3 Experimental Results

Identifying Comparative Sentence by Sentence Structure Templates

Sentence structure template is suitable for explicit comparative sentence identification, so we would firstly verify the coverage of the three kinds of templates mentioned in Sec.3.2. Our corpus totally contains 2,000 comparative sentences and 2,000 non-comparative sentences. In the experiment, we analyze the sentence structure for 2,000 comparative sentences and then calculate the coverage rate of three kinds of sentence structure templates. There are 1,890 explicit comparative sentences in 2,000 comparative sentences. It means that most comparative sentences are explicit. However, 1,735 explicit comparative sentences are matched by the three kinds of sentence structure templates in 1,890 explicit comparative sentences. The coverage rate, as high as 91.8%, shows that almost all the explicit comparative sentences are covered.

[5] http://www.zol.com.cn/

Through the experiment, we find that this approach only has a precision of 67.2% which isn't a satisfactory result. This is mainly because the templates are excessively broad and satisfy most of the sentence structure. So we need other auxiliary method to achieve a higher precision.

Identifying Comparative Sentence by Dependency Relation

During the experiment, we parse 2,000 comparative sentences with LTP and use the approach described in Sec.3.3 to calculate the similarity of dependency relation. As a result we achieve a high recall of 95.7% in 230 implicit comparative. It is clear that this approach is suitable for implicit comparative sentence identification. However, the approach has a low precision, and we still need other auxiliary method to achieve high precision.

Identifying Comparative Sentence by SVM

We take comparative words, statistical feature words and CSR as classification features of SVM, and orderly select the combination of features to do experiment. Where "CW" stands for comparative words, "SFW" stands for statistical feature words. The experimental results are shown in Table 3.

Table 3. The experimental results using SVM

Feature type	Precision (%)	Recall (%)	F-measure (%)
CW	68.2	73.1	70.9
SFW	67.6	73.5	70.4
CSR	**73.4**	**68.9**	**71.1**
CW+CSR	76.8	81.6	79.1
SFW+CSR	81.4	82.9	82.1
CW+SFW+CSR	**84.3**	**82.4**	**83.3**

Through analyzing the experimental results, we conclude that statistical feature word isn't obviously enough to represent comparative sentence. Similarly, only comparative word isn't effective. When CSR is involved in, the results are significantly improved. To some extent, it also proves that statistical features and rule-based features are complementary each other in mining comparative sentences, and also verifies that comparative sentence has important grammatical features. During the experiments, CSR as a single classification feature is the best and it proves that the main semantic information of comparative sentence is concentrated in the windows which take comparative word as the center of the windows with the range of 5. But the combination of CW, SFW and CSR is the best choice when we use SVM to identify comparative sentence. We achieve a high precision of 84.3% based on the statistical approach.

Identifying Comparative Sentence Based on the Combination of Rules and Statistics

In order to achieve high precision and recall, we use the combination containing sentence structure template (SS), the similarity of dependency relation (SDR) and SVM to identify comparative sentence. Firstly, the approach described in Sec.3.2 is used to identify explicit comparative sentence and the other described in Sec.3.3 is used to

Table 4. The experimental results based on the combination of rules and statistics

	Precision (%)	Recall (%)	F-measure (%)
SVM	84.3	82.4	83.3
SS+SVM	83.8	86.6	85.2
SDR+SVM	86.8	84.7	85.7
SS+SDR+SVM	**85.4**	**88.2**	**86.8**

identify implicit comparative sentence. Secondly, we use SVM to optimize identification results. The experimental process and results are shown in Table 4.

According to Table 4, the experimental results show that our approach based on the combination of rules (the broad extraction) and statistics (the accurate extraction) achieves a higher precision and recall. What's more, F-measure can reach up to 86.8%. Combining SS with SVM has a lower precision and a higher recall than SVM. Although combining SDR with SVM has the highest precision, the combination of SS, SDR and SVM has the highest F-measure than others. With the combination of them all, we ultimately obtain the satisfactory results of comparative sentence identification.

5 Conclusions and Future Works

In order to solve the problem of Chinese comparative sentence identification, we bring a brief summary and put forward a new solution. Further said that we propose a novel method of Chinese comparative sentence identification based on the combination of rules (the broad extraction first) and statistics (the accurate extraction later). Firstly we use comparative words and sentence structure templates to identify explicit comparative sentence. And then compute the similarity of dependency relation in comparative sentence to identify implicit comparative sentence. Finally in the result sets we apply CSR, comparative words and statistical feature words to optimize the identification results. The experiments with COAE 2013's test data show that our approach provides good performance. However, some issues need to be further studied. We will focus on the following questions in the future work:

(t) Use synonyms to expand the existing rule templates.
(u) Try to propose more general matching approach for dependency relation in comparative sentence.
(v) Need larger corpus to optimize comparative sentence identification algorithm.

References

1. Kawahara, D., Inui, K., Kurohashi, S.: Identifying contradictory and contrastive relations between statements to outline web information on a given topic. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 534–542. Association for Computational Linguistics (2010)

2. Jindal, N., Liu, B.: Identifying comparative sentences in text documents. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 244–251. ACM (2006)
3. Jindal, N., Liu, B.: Mining comparative sentences and relations. In: Proceedings of the National Conference on Artificial Intelligence. AAAI Press, MIT Press, Menlo Park, Cambridge (1999); 21(2), 1331 (2006)
4. Sun, J.T., Wang, X., Shen, D., et al.: CWS: A comparative web search system. In: Proceedings of the 15th International Conference on World Wide Web, pp. 467–476. ACM (2006)
5. Luo, G., Tang, C., Tian, Y.: Answering relationship queries on the web. In: Proceedings of the 16th International Conference on World Wide Web, pp. 561–570. ACM (2007)
6. Feldman, R., Fresko, M., Goldenberg, J., et al.: Extracting product comparisons from discussion boards. In: Seventh IEEE International Conference on Data Mining, ICDM 2007, pp. 469–474. IEEE (2007)
7. Li, S., Lin, C.Y., Song, Y.I., et al.: Comparable entity mining from comparative questions. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 650–658 (2010)
8. Huang, X.J., Wan, X.J., Yang, J.W., et al.: Learning to Identify Chinese Comparative Sentences. Journal of Chinese Information Processing 22(5), 30–38 (2008)
9. Huang, G.H., Yao, T.F., Liu, Q.S.: Mining Chinese Comparative Sentences and Relations Based on CRF algorithm. Application Research of Computers 27(6) (2010)
10. Zhang, R., Jin, Y.: Identification and Transformation of Comparative Sentences in Patent Chinese-English Machine Translation. In: 2012 International Conference on Asian Language Processing (IALP), pp. 217–220. IEEE (2012)
11. Che, J.: A brief analysis of comparative sentences in modern Chinese. Journal of Hubei Normal University (Philosophy and Social Science) 3 (2005)
12. Li, J.: The study of comparative sentence and comparison identification. Chongqing University (2011)
13. Song, R., Lin, H.-F., Chang, F.-Y.: Chinese Comparative Sentences Identification and Comparative Relations Extraction. Journal of Chinese Information Processing 23(2), 102–107 (2009)
14. Hu, B.-S., Wang, D.-L., Yu, G., et al.: An Answer Extraction Algorithm Based on Syntax Structure Feature Parsing and Classification. Chinese Journal of Computers 31(4), 662–676 (2008)
15. Liu, W., Yan, H.-L., Xiao, J.-G., et al.: Solution for Automatic Web Review Extraction. Journal of Software 21(12), 3220–3236 (2010)

Utility Enhancement for Privacy Preserving Health Data Publishing

Lengdong Wu, Hua He, and Osmar R. Zaïane

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{lengdong,hhe,zaiane}@cs.ualberta.ca

Abstract. In the medical field, we are amassing phenomenal amounts of data. This data is imperative in discovering patterns and trends to help improve healthcare. Yet the researchers cannot rejoice as the data cannot be easily shared, because health data custodians have the understandable ethical and legal responsibility to maintain the privacy of individuals. Many techniques of anonymization have been proposed to provide means of publishing data for research purposes without jeopardizing privacy. However, as flaws are discovered in these techniques, other more stringent methods are proposed. The strictness of the techniques is putting in question the utility of the data after severe anonymization. In this paper, we investigate several rigorous anonymization techniques with classification to evaluate the utility loss, and propose a framework to enhance the utility of anonymized data.

Keywords: Data Publishing, Privacy Preservation, Anonymization, SVM.

1 Introduction

Health research is central to the advancement of health care, which imperatively requires access to health data. However, health records are intrinsically identifiable, and control over the use of the data is necessary. When trying to publish health data, custodians of such data inevitable encounter hurdles relative to privacy[10]. To address such issues, a Privacy Rule was established in the United States which constitutes a comprehensive Federal protection for individuals' medical records and is known as *Health Insurance Portability and Accountability Act* (HIPAA)[12,13]. The legislation regulates health care groups, organizations or businesses, on how to use and disclose privacy data.

In recent years, several significant privacy preserving techniques have been proposed to protect individual's privacy when sharing the information. These techniques are progressively stricter as vulnerabilities in these algorithms are discovered. However, while such strict techniques prevent identification, they can significantly hinder the utility of the data for research purposes. In this paper we investigate several rigorous anonymization techniques with novel criterions based on a classification technique to evaluate the data utility. The remainder

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 311–322, 2013.

of the paper is organized as follows: in subsequent sections, we briefly introduce k-anonymity[1,4], l-diversity [3] and t-closeness [7] and their related limitations with tangible examples. We then present our utility evaluation methods on the anonymized data based on SVM, and the results of utility loss are analyzed. Given the sigificant utility loss, a privacy preservation with utility enhancement supervision framework is proposed. We present the implementation of the framework and algorithm with comparison experiment before concluding.

2 Privacy Preservation Technique

A typical health data table includes the basic personal information as well as their sensitive information such as diagnostic and treatment history records. All these attributes can be categorized into three classes [2]:

- *Identifier attributes*: a minimal collection of attributes that can explicitly identify individual records.
- *Sensitive attributes*: considered to be private.
- *Quasi-identifier (QI) attributes*: a minimal collection of attributes that can be linked with external information to re-identify individual records with high probability.

According to the HIPPA regulation, the removal of all identifier attributes is necessary. However, relinking attack [4,5] is a notorious attack on the de-identified tables by joining two tables having common quasi-identifier attributes. For example, based on the statistics, approximately 97% of 54,805 voters in the Cambridge, U.S. can be uniquely identified on the basis of full combination of the zip-code, gender and birthday attributes; 87% can be identified with the combination of only 5-digit ZIP-code, gender and birthday; and another 69% uniquely with the ZIP-code and birthday [2]. This result reveals a serious privacy preservation problem and shows a high possibility of re-identifying the de-identified table under the re-linking attack.

2.1 *k*-Anonymity beyond De-identification

***k*-Anonymity Principle.** The simple identifier removal process cannot guarantee the anonymity of the published data due to its potential leakage on *QI* attributes. The k-anonymity technique is designed to avoid re-linking attacks through generalizing the *QI* attribute values. For each *QI* attribute, a tree-structured domain generalization hierarchy is maintained, in which the node in higher levels contains more generalized information. Given this hierarchy, the specific values in the original table can be replaced by the more general values in higher level nodes of the hierarchy. Records with the same generalized value are gathered into an equivalence class[2], thus the re-linking attack cannot distinguish a certain individual from other records in the certain equivalence class.

A table satisfies k-anonymity principle if at least k indistinct records exist in each equivalence class. For instance, Table 1 satisfies 3-anonymity.

Attacks on *k*-Anonymity. It is quite common for k-anonymity to generate equivalence classes with same values of sensitive attributes, especially when certain sensitive attributes have high frequent values. For example, in Table 1, an adversary can easily know that individuals in the second equivalence class suffer from Gastric Ulcer. Although the equivalence class decreases the possibility of identifying individual, the sensitive attributes can provide auxiliary clew, which can be utilized by homogeneity attacks[3]. Background attack[14] uses some background knowledge to obtain privacy information on the k-anonymity tables. Again, in Table 1, suppose an adversary knows that an individual in the first equivalence class has a certain cancer, this fact as background knowledge can assure the adversary that this individual has Stomach Cancer.

Table 1. k-Anonymity Health Table (k=3)

sq	ZIP-code	Age	Sex	Disease
1	476**	6*	*	Gastritis
2	476**	6*	*	Gastric Ulcer
3	476**	6*	*	Stomach Cancer
4	97***	5*	F	Gastric Ulcer
5	97***	5*	F	Gastric Ulcer
6	97***	5*	F	Gastric Ulcer

2.2 *l*-Diversity beyond *k*-Anonymity

***l*-Diversity Principle.** To deal with the defects of k-anonymity, l-diversity requires that the sensitive attribute values in each equivalence class should be as diverse as possible, requiring at least l well-represented sensitive attribute values. The requirement of l well-represented values of sensitive attributes adds an extra protection layer over k-anonymity. When a table satisfies l-diversity, the adversary who breaches the k-anonymity, still needs to exclude the (l-1) possible sensitive values. The larger the parameter l, the more protection it provides.

Attacks on *l*-Diversity. However, the requirement of l-diversity on well-represented values cannot really ensure the real diversity for sensitive attributes. For example in Table 1, "Gastric Ulcer", "Gastritis" and "Stomach Cancer" are all stomach related, then the adversary could know that the individuals in the first equivalence class must have a problem with the stomach. Similarity attack [6] and skewness attack [7] are two typical attacks on such semantic leaks in sensitive values. The breach will be serious when the number of sensitive attribute categories is small.

2.3 t-Closeness beyond l-Diversity

Rather than simply making sensitive attribute values numerically diverse, t-closeness [7] makes the sensitive values semantically diverse. The t-closeness requires the distribution of sensitive values in each equivalence class close to the overall distribution of the whole table.

3 Utility Loss Evaluation

3.1 Utility Loss Measures

The three important privacy preservation processes, k-anonymity, l-diversity and t-closeness are effective in protecting data privacy. However, there is a risk that they lower the utility of the data in the context of health research, such as building classifiers for automated diagnostic, treatment recommendation or other relevant applications requiring machine learning. Therefore, the balance between the data utility for scientific research and the privacy preservation for health data is of paramount importance; at least, reducing the loss as much as possible while keeping the same level of privacy, is imperative.

To capture data utility, some criteria measure the utility loss that is incurred by generalization based on generalization hierarchies, such as Discernability Measure (DM) [1], Utility Measure (UM) [17], Relative Error (RE) [18], Normalized Certainty Penalty (NCP) [16] etc. DM and RE is calculated based on the number of generalized group and suppressed group that overlap with the original data. NCP and UM are expressed as the weighted average of the information loss, which are penalized based on the number of ascendants in the hierarchy. Some recently proposed measures, such as multiple level mining loss [15], express utility based on how well anonymized data supports frequent itemset mining. However, all these measures are essentially evaluating the information loss of generalized items via certain penalization function based on the number of ascendants achieved in the hierarchy. A measure that can be used in the absence of hierarchies and captures the utility loss incurred by generalization is more preferred by practical application scenarios.

Machine learning applications can utilize the analysis and intelligent interpretation of large data in order to provide actionable knowledge based on the data for human decision support or automatic decision making. Support Vector Machine (SVM) is one of the effective machine learning algorithm. The standard SVM takes a set of input, each of which belonging to one of several categories; then builds a model of hyperplane separating the data space through the learning process to predict whether a new test example falls into one category or another. In this section, we are particularly interested in evaluating and discussing the utility loss induced by privacy protection via the use of Support Vector Machine (SVM), and examine the utility value through the measure of accuracy after anonymization.

3.2 Datasets and Experimental Setup

We use two census-based datasets, the Adult dataset, which is originally from the US census bureau database, and the IPUMS dataset from the historical census project at the University of Minnesota. Both datasets, available from the UCI Machine Learning repository[1], have been extensively used by recent privacy preservation studies [3,7]. In the Adult dataset, we choose attribute set including *age*, *workclass*, *education*, *gender*, *race*, *marriage*, and *country* as QI attributes, and use the salary class as the sensitive attribute. In the IPUMS dataset, QI attribute set includes *sex*, *relationship*, *race*, *birthplace*, *children number*, *education attainment*, *weeks worked last year*, and use the *wage* class as the sensitive attribute. We remove all records with missing values. Our experiments use the following parameters: $k = 4$ for k-anonymity, $k = 4$ and $l = 2$ for l-diversity, $k = 4$ and $t = 0.2$ for t-closeness. Those settings are commonly applied in practice [3,5,7], which are regarded to be not too strict to make the output data completely unusable.

We use the LibSVM toolkit[8] to run the SVM classification algorithm, and apply the same SVM parameters for all experiments. The datasets are divided into the training and test sets randomly in three fold cross validation sets: one third of each set is used as test data while the other two thirds are used for training. In our first experiment, we use SVM on the original dataset, so that all information in the *QI* attributes can be fully utilized for SVM classification. We then apply SVM on the anonymized data by k-anonymity, l-diversity and t-closeness separately. By comparing the classification results, we can evaluate to what degree the anonymized data could lose utility and we examine its loss value.

3.3 Utility Loss Results

Table 2 presents the comparisons of the accuracies of correctly classified records by SVM on the Adult dataset. Significant drops, 25% in sensitivity and 21% in specificity, can be observed due to the inadvertent obfuscation of pertinent information necessary for building the classification model. In Table 2, one might expect the classification results to have lower accuracies for l-diversity and t-closeness compared to k-anonymity; however, the results are quite similar. This is due to the fact that k-anonymity already produces significant information loss, and l-diversity in each equivalent class is already established. Table 3 shows the comparison based on the IPUMS data, where there is a noticeable drop when using l-diversity and t-closeness as compared to k-anonymity. This shows an additional utility loss beyond what k-anonymity can have already done.

Based on the experimental results on these datasets, we can conclude that the utility value of data, after the anonymization by k-anonymity, l-diversity and t-closeness, is significantly jeopardized due to the strictness of those privacy preservation mechanisms.

[1] `http://archive.ics.uci.edu/ml/machine-learning-databases/`

Table 2. Experiment Results with Adult Dataset

	SENSITIVITY/RECALL	*SPECIFICITY*	*ACCURACY*	*PRECISION*	*F-MEASURE*
Original	88.0%	76.3%	82.1%	78.6%	83.3%
k-Anonymity	63.6%	55.6%	60.3%	58.9%	61.4%
l-diversity	62.3%	53.5%	59.2%	56.3%	59.4%
***t*-closeness**	62.7%	53.2%	59.7%	54.1%	58.5%

Table 3. Experiment Results with IPUMS Data

	SENSITIVITY/RECALL	*SPECIFICITY*	*ACCURACY*	*PRECISION*	*F-MEASURE*
Original	79.6%	76.5%	77.9%	77.2%	78.8%
k-Anonymity	64.5%	61.5%	63.3%	62.1%	64.8%
l-diversity	57.6%	56.2%	57.5%	58.8%	58.9%
***t*-closeness**	53.6%	55.1%	54.2%	55.4%	54.5%

4 Privacy Preservation with Utility Supervision

4.1 Utility Enhancement Supervision Framework

To minimize the utility loss of these privacy preserving techniques, partition-based and cluster-based anonymization algorithms have been proposed recently. The partition-based anonymization treats a record projected over QI attributes as a multi-dimensional point. A subspace that contains at least k points forms a k-anonymous group [18]. The main idea of clustering-based anonymization is to create clusters containing at least k records in each cluster separately [16]. Fung et al. [19] presented an effective top-down approach by introducing multiple virtual identifiers for utilizing information and privacy-guided specialization. However, the partitioned-based anonymization selects the attribute with the largest domain for efficiency and top-down specialization chooses the attribute with best pre-defined scoring ranking. These genetic evolution and top-down generalization algorithms do not produce any progressive attribute selection process which determines a desired balance of privacy and accuracy.

We introduce the utility enhancement supervision in the attribute selection process. The insight of our proposal is based on the acknowledgement that the anonymization process unquestionably damages the potential data correlation between the QI attributes and the sensitive attributes; and the higher generalization hierarchy is achieved the more correlation is lost. Since any prior knowledge is unknown about the class related features for QI attributes, there probably exist, among the numerous QI attributes, some that have poor correlation or no correlation with sensitive attributes. These superfluous QI attributes are definitely ideal for generalization without losing any utility. More generally, QI attributes that are less correlated with the sensitive attribute are better candidates for generalization of anonymity than others. The less the attributes with strong correlation are generalized, the more utility will be preserved for anonymization.

Hence, the utility enhancement supervision is established to produce such an order of QI attribute candidates for generalization.

Figure 1 illustrates our framework of privacy preservation with utility enhancement supervision. The process is divided into four stages:

- Stage 1. Sample data extraction. De-identified dataset is submitted to D and sample data D_0 is randomly extracted from D for evaluation purpose.
- Stage 2. Anonymization candidates order. Given the randomly selected sample dataset D_0, SVM utility evaluation is applied to produce the partial order of correlation of QI attributes.
- Stage 3. Attribute generalization. Optimal attributes are chose based on the partial order to be generalized according to each own generalization hierarchy.
- Stage 4. The anonymized dataset D' is verified according to anonymity principles. If all equivalent classes satisfy all requirements of the specified principle, D' is ready for publishing.

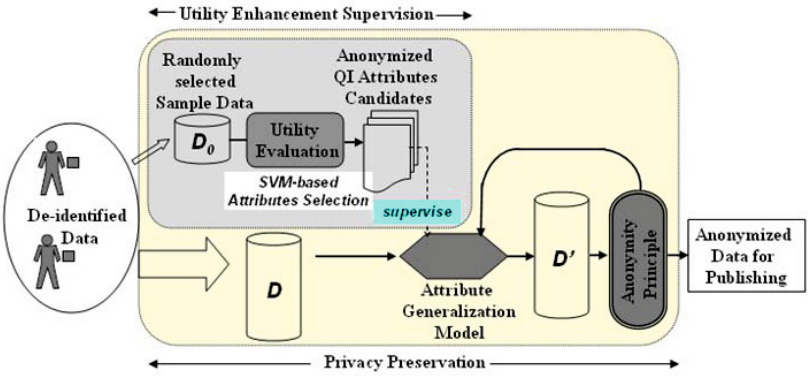


Fig. 1. Privacy Preservation with Utility Supervision Framework

4.2 Privacy Preservation with Utility Supervision Algorithm

To produce an appropriate anonymization algorithm with utility supervision, we need to solve the following issues:

- define the standard for comparison which is essential for the partial order.
- devise an efficient algorithm to generate the partial order.
- select the optimal attribute candidates for generalization based on the utility order

For this purpose, we continue to adopt the utility evaluation based on SVM and the F-measure value for SVM-based classifier cross validation is used as the

criteria for comparison. We use the notation $F(S)$ to indicate the F-measure value for cross validation with attributes set S.

To generate the partial order of candidates, the simplest way is to compare all possible combinations. However, the number of combinations grows exponentially as the number of attributes increases, thus the brute-force solution might not always be practical. Thus we use sequential backward selection (SBS), to achieve affordable search performance. We assume the original QI attributes set is X_s. Each time one attribute ξ is removed, and SVM classifier is done based on attributes $(X - \xi)$ obtaining $F(X - \xi)$. The attribute $\hat{\xi}$ having the maximum F value implies that the left attributes $(X - \hat{\xi})$ can best preserve utility, thus $\hat{\xi}$ is removed. The removal procedure is repeated until the attribute set is empty with a utility tree established.

To extract the attribute candidates, we first find the maximum $F(X')$ value in the whole tree, then attributes existing in $(X - X')$ will be chosen for generalization. In the case that these first-batch candidate attributes are all generalized to their highest level in the generalization hierarchy and the anonymization constraints are still not satisfied, another batch of candidates need to be selected for further generalization. For this purpose, the maximum $F(X'')$ value is searched in the subtree whose root is X'. Attributes in $(X' - X'')$ will form a new group of attributes for generalization. The procedure of search, selection, generalization and check is executed repeatedly until a certain anonymization principle is achieved. Algorithm 1 demonstrates the details for the procedure.

For example, we assume there are six QI attributes $X = \{A, B, C, D, E, F\}$, as illustrated in Figure 2. After removing each attribute and executing a classifier cross validation, we find that $X_3 = X - C = \{A, B, D, E, F\}$ obtains the highest F value. Thus in the next round of tree building, we only start from X_3 rather than considering other sibling nodes. With the same manner for $X_{34} = \{A, B, E, F\}$, $X_{342} = \{A, E, F\}$, and $X_{3425} = \{A, F\}$, the utility tree can be established. To select attribute candidates, the maximum F value is achieved by $F(X_3)$ with attribute set $\{A, B, D, E, F\}$. Thus, QI attribute $\{C\}$ is first chosen to be generalized. In the subtree of X_3, X_{342} with attribute set $\{A, E, F\}$ has the highest F value. $\{B, D\}$ will be the candidates for generalization. Repeatedly, $\{A, E\}$ will be selected as next group of candidates.

4.3 Experiment Evaluation

The experiment is based on the same census-based datasets, the Adult dataset and the IPUMS dataset, and the same QI attributes set are chosen as introduced in Section 3.2. We compare the utility loss by global recoding and local recoding implementation [15,16] and test with the common configurations for k, l and t as described in Section 3.2. The global recoding can be described by a group of functions $\phi_i : D_{X_i} \rightarrow D'$ for each attribute X_i of the Quasi-identifier. The anonymization is obtained by applying each ϕ_i to the values of X_i in each tuple of D. The global recoding generalizes the attribute for a group of records in the table for efficiency. In our implementation, the data space is partitioned into a

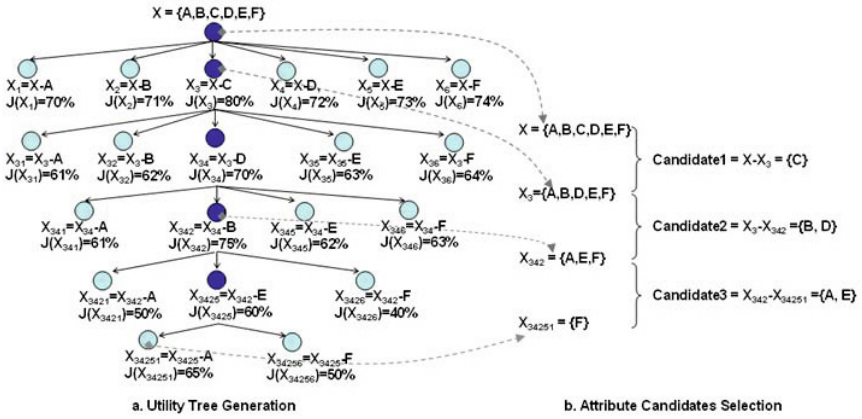


Fig. 2. Example of Privacy Preservation with Utility Supervision Algorithm

set of non-overlapping regions and the algorithm maps all records in a region to the same generalization hierarchy level. When checking whether anonymity principle is satisfied, a single SQL query can be established, for example, "SELECT *race*, *gender*, *age*, *count(*)* FROM *Table* GROUP BY *race*, *gender*, *age* HAVING *count(*)*> k". Alternatively, the local recoding is described by a function $\phi : D_{X_1} \times D_{X_2} \times ... \times D_{X_n} \rightarrow D'$, which recodes the domain of value vectors associated with the set of Quasi-identifier attributes. Under this definition, the anonymization is obtained by applying ϕ to the vector of Quasi-identifier values in each tuple of D. We implemented the local recoding by generalizing the Quasi-identifier attribute to a higher level only for the distinct individual record that does not achieve the anonymity constraints rather than all records. To check the satisfaction of anonymity constraints, we introduce an *equivalence class id*. The record satisfying the constraints is assigned with such a class id, indicating that the record belongs to the corresponding equivalence class after generalization. When each record in the table has a valid class id, the table is considered to be anonymized successfully.

Based on the algorithm 1, in the Adult Dataset, we obtained the attribute set partial order as: F(*age*, *workclass*, *education*, *country*)=84.4%, F(*workclass*, *education*)=78.2%. Thus, generalization is done firstly on attribute set {*race*, *marriage*, *gender*}, and after all these attributes have reached the highest level in the hierarchy, attribute set {*age*, *country*} is generalized. In the IPUMS dataset, attribute set partial order is calculated as: F(*sex*, *race*, *children number*, *education attainment*, *occupation*, *weeks worked last year*)=79.2%, F(*sex*, *children number*, *education attainment*, *occupation*, *weeks worked last year*)=73.4%, F(*education attainment*, *occupation*, *weeks worked last year*)=70.6%. Accordingly, attribute candidate sets are generalized based on the order: {*relationship*, *birthplace*}, {*race*}, {*sex*, *children number*}.

Algorithm 1. Anonymization with Utility Supervision

Input: Private de-identified Table, QI($\xi_1, ..., \xi_n$), Anonymity constraints, Domain generalization hierarchy $DGH_{\xi_i}, i \in [1, ..., n]$

Output: Anonymized Publishable Table containing a generalization over QI with respect to Anonymity principle

`/* Step1. Generate utility tree of QI attributes based on SBS */`
initial selected attributes set $X_s \leftarrow$ QI($\xi_1, ..., \xi_n$) ;
initial root node $\leftarrow F(X_s)$;
repeat
  foreach $\xi_i \in X_s$ **do** remove one attribute from the X_s
    `/* Use SVM-based classifier on randomly selected sample data for cross validation */`
    $F_i \leftarrow F(X_s - \xi_i)$;
    $F(X_s)$.child node $\leftarrow F_i$;
  end
  `/* Find such attribute` ξ_k `that` $F(X_s - \xi_k)$ `is the maximum */`
  $F_k \leftarrow Max(F_i)$, $i \in [1, ...s]$;
  $X_s \leftarrow X_s - \xi_k$;
until $X_s = \perp$;
`/* Step2. Search for candidates for generalization in utility tree */`
Initial root node $X_s \leftarrow X$;
repeat
  Search Tree(X_s) $\rightarrow X' : F(X')$ is maximum ;
  repeat
    `/* Step3. Generalize attribute candidates */`
    Select attribute set $< X_s - X' >$ for generalization ;
    Build hierarchy vector $< DGH_{\xi_i}+1 >$, $\xi_i \in < X_s - X' >$;
    Replace the new hierarchy vector to $< X_s - X' >$ in equivalent class;
    `/* Check data is anonymized successfully for publishing. */`
    if *(Anonymity constraints are satisfied by all equivalent classes)* **then**
      return;
    end
  until *Highest level in each* DGH_{ξ_i};
  `/* Start a new round candidates search in the Tree(`X'`) for generalization */`
  $X_s \leftarrow X'$;
until *Anonymity constraints are achieved*;

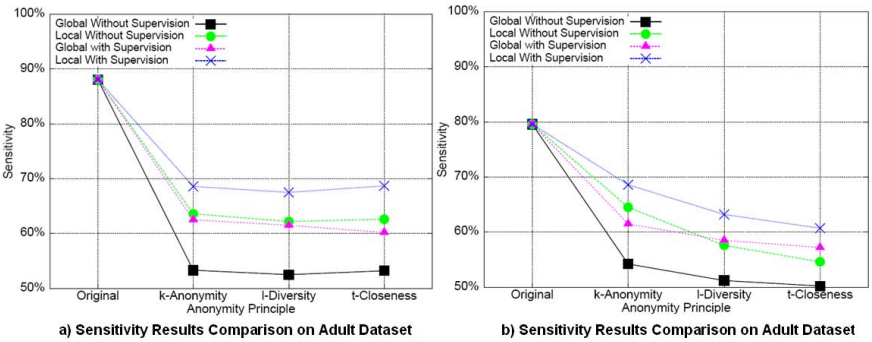


Fig. 3. Performance Comparison of Privacy Preservation with vs. without Supervision

Figure 3 shows that there is a significant increase in terms of sensitivity and specificity between anonymization with supervision and anonymization without supervision on both datasets. On the Adult dataset, we get an accuracy about 7% higher for k-anonymity and l-diversity principle, and 5% higher for t-closeness. Such significant rises are due to the deliberate retainment of pertinent attribute information necessary for building the classification utility model. Comparison on the IPUMS data shows the accuracy for the classifier can be improved even more when using l-diversity and t-closeness principle than with k-anonymity. This is because l-diversity and t-closeness, being stricter than k-anonymity, they necessarily require further generalization on additional attributes. Imposing restrictions or guidance on the attributes being generalized can reduce the risk that pertinent information contained by correlative attributes is jeopardized. Based on the experimental results on these datasets, we can conclude that our proposed privacy preservation algorithm with utility supervision can significantly increase the utility of privacy preservation mechanisms.

5 Conclusion

In this paper we examined the issue of utility of health data after the anonymization process and put forward the necessity of finding a trade-off between privacy protection and utility of data. We describe three important and recently proposed privacy preservation techniques, k-anonymity, l-diversity and t-closeness, and present the limitations of each technique. By using SVM to evaluate the utility loss, we show that the privacy preservation technique implementation we have at our disposal today can significantly jeopardize the data utility due to the obliteration of pertinent information. Protecting the privacy of patients is central. Using the wealth of health data we are collecting to improve healthcare, is also essential. To enhance the utility of the data we put forward the privacy preservation with utility enhancement supervision framework. With this framework, the anonymized data is able to preserve the data utitily as well as protect the privacy of sensitive information.

References

1. Bayardo, R.J., Agrawal, R.: Data Privacy through Optimal k-Anonymization. In: Proceedings ICDE 2005, pp. 217–228. IEEE (2005)
2. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems 10(5), 557–570 (2002)
3. Machanavajjhala, A., et al.: L-diversity: privacy beyond k-anonymity. In: TKDD 2007, vol. 1(1). ACM (2007)
4. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems 10(6), 571–588 (2002)
5. LeFevre, K., De Witt, D., Ramakrishnan, R.: Incognito: Efficient full-domain k-anonymity. In: Proceedings of SIGMOD 2005, pp. 49–60. ACM (2005)
6. Truta, T.M., Vinay, B.: Privacy protection: p-sensitive k-anonymity property. In: Proceedings of ICDE 2006. IEEE Computer Society (2006)
7. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: privacy beyond k-anonymity and l-diversity. In: 23rd International Conference on Data Engineering, pp. 106–115. IEEE Computer Society (2007)
8. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011)
9. Zhong, S., Yang, Z., Wright, R.N.: Privacy-enhancing k-anonymization of customer data. In: Proceedings of the 24th ACM SIGMOD Symposium on Principles of Database Systems, pp. 139–147. ACM, New York (2005)
10. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Trans. on Knowl. and Data Eng. 13(6), 1010–1027 (2001)
11. Kasiviswanathan, S.P., Rudelson, M., Smith, A., Ullman, J.: The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, Cambridge, Massachusetts, USA, pp. 775–784 (2010)
12. Notice of Addresses for Submission of HIPAA Health Information Privacy Complaints Federal Register 68(54), March 20 (2003)
13. Sweeney, L.: Weaving technology and policy together to maintain confidentiality. Journal of Law, Medicine and Ethics 25, 98–110 (2000)
14. Wong, R., Li, J., Fu, A., Wang, K.: (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 754–759 (2006)
15. Terrovitis, M., Mamoulis, N., Kalnis, P.: Local and global recoding methods for anonymizing set-valued data. The International Journal on Very Large Databases 20(1), 83–106 (2011)
16. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.C.: Utility-based anonymization using local recoding. In: SIGKDD 2006, pp. 785–790. ACM (2006)
17. Loukides, G., Shao, J.: Capturing data usefulness and privacy protection in k-anonymisation. In: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 370–374. ACM (2007)
18. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: ICDE 2006, pp. 25–36. IEEE (2006)
19. Fung, B.C., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation. In: ICDE 2005, pp. 205–216. IEEE (2005)

Optimizing Placement of Mix Zones to Preserve Users' Privacy for Continuous Query Services in Road Networks

Kamenyi Domenic M., Yong Wang, Fengli Zhang,
Yankson Gustav, Daniel Adu-Gyamfi, and Nkatha Dorothy

School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, P.R. China
{cla,fzhang}@uestc.edu.cn,
dkamenyi@yahoo.co.uk

Abstract. Location Based Services (LBS) are becoming very popular with today's users. Some of these LBS require users to continuously send requests for services. This lead to leakages of both location and query contents to malicious adversaries. Further, if users are constrained by the nature of the road networks, an adversary can follow their path trajectory with ease. Most of the current privacy preserving solutions focus on temporal and spatial cloaking based methods to protect users' location privacy. However, these solutions are vulnerable when subjected to continuous query environments. In this paper, we propose an optimized solution that preserves privacy for users' trajectory for continuous LBS queries in road networks. First, we deploy a trusted third party architecture to provide anonymity for users as they use LBS services. Second, we utilize mix zone techniques and design two algorithms. The first algorithm, Abstraction Graph (AG), selects a sample of mix zones that satisfy the user desired privacy level under the acceptable service availability condition. The second algorithm, Optimized Decision Graph (ODG), utilizes the generated graph to find an optimal solution for the placement of mix zones through decomposition, chunking and replacement strategies. Finally, we analyze the capability of our algorithms to withstand attacks prone to mix zones and carry out experiments to verify this. The experiments results show that our Algorithms preserve privacy for users based on their privacy and service availability conditions.

Keywords: Location-based Services (LBSs), Privacy Preservation, User Trajectory, Continuous Query, Decision Graphs.

1 Introduction

In recent time, there has been widespread use of LBS through mobile technology with GPS capability [9]. There are two types of privacy concerns in LBS; *location privacy* where the aim is to protect sensitive location from being linked to a specific user and *query privacy* where query is protected from being linked to a user.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 323–335, 2013.

Further, query can either be *snapshot* or *continuous* [11,10]. Our concern is to preserve privacy for continuous query that is difficult to achieve. For example, in continuous query, an adversary can follow users' trajectory over Euclidean space and break their security. Worse still, an attacker can easily use the constrained road network setup to follow users' trajectory with ease. However, privacy can be achieved by use of mix-zone frameworks [11]. When users simultaneously enter an intersection (designated as a mix-zone), they change to a new unused pseudonym. In addition, while in a mix zone, they do not send their location information to any location-based application. An adversary tracking a user will not distinguish users entering a mix-zone with those coming out of it.

To illustrate how mix zone work, take an example shown in Fig. 1. vehicle numbers 1, 2 and 3 enter an intersection using road segment A. Vehicle numbers 4, 5 and 6 enter the same intersection through road segment C. Vehicles entering the junction from A can exit through road segment B, C or D. Likewise, vehicles accessing the junction from C can exit either through A, B or D. The adversary cannot distinguish or even pinpoint correctly vehicles leaving this junction.

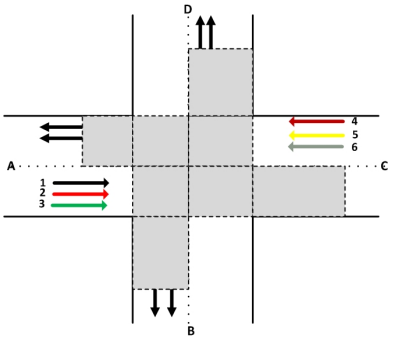


Fig. 1. Non-Rectangular Mix Zone

However, under certain conditions, users' trajectory may be exposed that may lead to three types of attacks. First, *timing attack* [10] occur when a group of users enter and leave a mix-zone within a small difference in their time. Second, *transition attack* [10] occur when an adversary utilize the users' transition of turning (right, left or going straight), if the number entering an intersection simultaneously is not large enough. Finally, *continuous query attack* occur where the speed and transition of a user requesting for continuous service is different from other users. Attacker may utilize these phenomenon to infer the target user.

In this paper, we offer our contributions towards finding a solution to these privacy exposing attacks. First, we deploy a trusted third party architecture of anonymizing servers to preserve users' privacy while accessing LBS as well as maintaining desired service availability. Second, we use mix zone approach to design two algorithms. The first algorithm (Abstraction Graph - AG) select a sample of mix zones that satisfy the user desired privacy and service levels, while the second algorithm (Optimized Decision Graph - ODG) finds an optimal solution for the placement of mix zones through decomposition, chunking and

replacement strategies. Finally, we analyze the capability of our algorithms to withstand mix zones attacks and carry out experiments to verify this.

The rest of the paper is organized as follows: In section 2, we present related work followed by system design in section 3. A detailed explanation of the proposed algorithms is given in section 4. Section 5 presents Security and Privacy Analysis with experiments and evaluations in section 6. Finally, in section 7, we conclude with a proposal for future work.

2 Related Work

We categorize related work into two parts. The first part explores recent research on privacy preservation in road networks and the second part considers privacy preservation in road network using mix zone approach.

2.1 Privacy Preservation Techniques in Road Networks

In recent times several techniques have been proposed on how to preserve privacy in road network. One of these techniques is Ting Wangs' [14] et al. *X-Star*. They regard the *attack resilience* and the *query-processing cost* as two critical measures for designing location privatization solutions. However, *X-Star* Framework incurs low anonymization success rate and it's computation cost is quite high. Al-Amin Hossain [4] et al. proposed Hilbert-order based star network expansion cloaking algorithm (*H-Star*) that guarantees K-anonymity under the strict reciprocity condition and increases anonymization success rate by reducing computation overhead. However, this framework does not support continuous location based queries. Further, Joseph T. Meyerowitz [8] et al. developed *CacheCloak*, a system that anonymize a user by camouflaging their current location with various predicted paths. They extended the idea of *path confusion* and *predictive path confusion* to enable *caching* that generates a predicted path for the user. They considered applications that can operate using user's location. However, some applications require more than just the user's current location to operate.

Chi-Yin Chow [2] et al. noted that applying the above techniques directly to the road network environment lead to privacy leakage and inefficient query processing. They proposed "*query-aware*" algorithm designed specifically for the road network environment that takes into account the query execution cost at a database server and the query quality during the location anonymization process. However, the proposed algorithm only works with snapshot locations. Further, Chi-Yin Chow [1] et al. gave a survey on the state-of-the-art privacy-preserving techniques in snapshot and continuous LBS. They noted that protecting user location privacy for continuous LBS is more challenging than snapshot LBS.

2.2 Privacy Preservation Using Mix Zones in Road Networks

One of the recent techniques for preserving privacy in road network is mix zone approach. Balaji Palanisamy [11] et al. proposed *MobiMix* framework that

protect location privacy of mobile users on road networks. They provided the formal analysis on the vulnerabilities of directly applying theoretical rectangle mix-zones to road networks in terms of anonymization effectiveness and attack resilience. They proposed use of non-rectangular mix zones. Further, Kai-Ting Yang [15] et al. argues that the concept of continuous location privacy should be transferred to users' path privacy, which are consecutive road segments that need to be protected and proposed a novel M-cut requirement to achieve this.

Xinxin Liu [7] et al. investigated a new form of privacy attack to the LBS system where an adversary reveals user's true identity and complete moving trajectory with the aid of *side information*. They proposed a new metric to quantify the system's resilience to such attacks, and suggested using multiple mix zones as a cost constrained optimization problem. Further, Murtuza Jadliwala [5] et al. studied the problem of determining an optimal set of mix-zones such that the degree of mixing in the network is maximized and the overall network-wide mixing cost is minimized. They modeled the optimal mixing problem as a generalization of the vertex cover problem. In the *Mix Cover*, as it was called, they proposed three approximation algorithms and two heuristics. However, the problem of Continuous Query (C-Q) attacks [10] was not tackled by above research.

3 System Design

3.1 Designing Goals

First, we adopt trusted third party architecture to achieve anonymity. Second, we introduce two terms; 1) *demand* - d for a road segment that represents the average number of users in a road segment (traffic capacity on a segment), and 2) *cost* - c at each vertex that represents the average cost (per user) of mix-zone deployment at that intersection (intersection mixing cost). We use *cost* to select a sample of mix zones to act as the population of promising solution. We then use *demand* to maximize on mix zones entropy in order to confuse the adversary. We optimize this solution using decomposition, chunking and replacement strategies [13]. Finally, we deal with mix zone attacks [11] by deploying an Optimized Decision Graph that achieves greater anonymity and service availability.

3.2 System Architecture

The proposed architecture uses secure communication channel composing of mobile users, anonymizing server and LBS as in Fig. 2. AS consists of 3 components: 1) **Optimizing Decision Engine** that use hierarchical Bayesian Optimization Algorithm - *hBOA* to generate Graphs from road networks that satisfy client service availability and feeling safe conditions. The generated graph and query content are then forwarded to LBS for service. 2) **Repository** that stores generated Graphs that are tagged with time, date and the month they were generated. 3) **Result Refiner** that is responsible for refining the accurate result from candidate ones sent by the LBS according to the knowledge of the client's position.

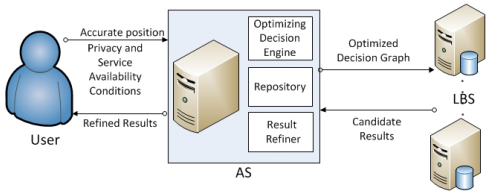


Fig. 2. System Architecture

A user will submit to AS, its position, query content, privacy profile and service availability condition. AS will retrieve and forward to LBS, the relevant Optimized Decision Graph and query content. Based on this information, LBS will figure out candidate results and return them to AS. AS will calculate refined results and send them to client. Updates to AS will be in form of: 1) privacy conditions, 2) new service availability conditions, and 3) new areas to be visited. AS will then use the new updates to generate Optimized Decision Graphs for storage. For a continuous query, AS will continue to receive location information updates from the client until the expiry of the query. Based on the updated client location, AS will continue to retrieve relevant decision graphs that preserve users privacy and update LBS. Further, we treat LBS as un-trusted. We discuss how to deal with attacks by malicious adversary in Section 5.

4 Algorithms

4.1 Preliminaries

The problem of optimizing mix zone placement is NP-Hard as discussed in [7,3]. However, we can have a Linear Programming relaxation of this problem. If LP relaxation has an integral solution then that can also be a solution. For example, let assume that we have a binary decision variable for each edge e and its corresponding vertex v which indicates whether the vertex v is included in the selected population of promising solutions for edge e or not. That is, if included we represent it with binary 1 and if not a 0. Let d_v be the decision variable indicating the demand on a road segment $e\ \varepsilon\ E$ and c_v indicating the cost of each vertex $v\ \varepsilon\ V$. The linear Programming of this problem can be presented as:

$$\begin{gathered}\text{Min } \textstyle\sum_{v\epsilon V}\ d_v.c_v \\ \text{Subject to: } d_e^v du + d_e^u dv \geq d_e^u d_e^v, \forall e \equiv (u,v)\epsilon E \\ d_v \geq 0, \forall v\epsilon V\end{gathered}$$

To solve this problem, we model the location map as a directed weighted graph *G(V;E;d;c)*, where V is the set of vertices, E is the set of road segments, d is the *demand* on a road segment given by the average number of users in that road segment, and c is the *cost* at each vertex given by the average cost (per user) of mix-zone deployment at that vertex. Further, two vertices are said to be pairwise

connected when there is at least one path connecting them. We introduce a mix-zone to break pairwise connectivity in order to achieve anonymity.

First, We classify the road network by clustering according to traffic in road segments. For example, segments in major roads carry more traffic than roads feeding major roads. We generate a Decision Graph with the top most layer having segments from major roads, followed by other smaller roads. Second, we find an optimized solution on the placement of mix zones to achieve the desired privacy for users as well as acceptable service availability. The ideal situation is to place mix zones in every road intersection. However, this is not practical. The actual intersection cost resulting from a road segment with intersection at v depends on the intersection mixing cost c and the traffic capacity d. We perform a selection that minimizes the intersection mixing cost as well as maximizing on traffic capacity on road segment that affect entropy (to be introduced later).

To achieve optimized solution we propose to use Hierarchical Bayesian Optimization Algorithm *(hBOA)* [13] that captures hierarchical nature of our problem at hand. In this case, higher layers captures traffic on major roads and lower layers that of streets. *hBOA* accomplishes 3 steps; 1) *Decomposition* - in each level the problem is decomposed properly by identifying most important interactions between the problem variables and modeling them appropriately, 2) *Chunking* - partial solutions are represented at each level compactly to enable the algorithm to effectively process partial solutions of large order, and 3) *Diversity maintenance* - alternative partial solutions are preserved until it becomes clear which partial solutions may be eliminated (*niching*). To ensure *decomposition* and *chunking*, we use decision graphs to build Bayesian networks from the selected population of promising solutions and capture local structures to represent parameters of the learned networks. To ensure *diversity maintenance*, we use *restricted tournament replacement (RTR)* [13] to satisfy *niching*.

For example, consider a binary variable X_1 which is conditioned on 4 other binary variables denoted by X_2, X_3, X_4 and X_5. A fully conditional probabilities table for X_1 containing 16 entries (2^4) is generated and after proper decomposition and chunking, we get a Decision Tree and Graph as in Fig. 3. A sequence of splits and merges are done without losing the original meaning and a further reduction on storage space is achieved. We only store values (0.25, 0.45, 0.75).

X_2	X_3	X_4	X_5	$P(X_1 = 0 \mid X_2, X_3, X_4, X_5)$
0	0	0	0	0.75
0	0	0	1	0.45
0	0	1	0	0.45
0	1	0	1	0.45
0	0	1	0	0.45
0	0	1	1	0.45
0	1	0	0	0.45
0	1	1	1	0.45
1	1	1	0	0.25
1	1	0	1	0.25
1	0	1	0	0.25
1	0	0	1	0.25
1	0	0	0	0.25
1	1	0	1	0.25
1	1	1	0	0.25
1	1	1	1	0.25

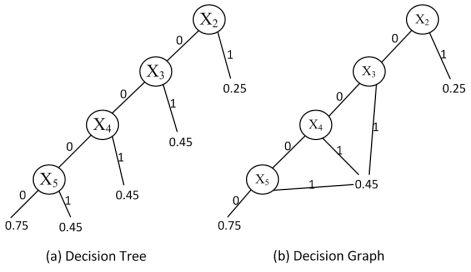


Fig. 3. Decision Tree and Decision Graph that encodes the probabilities in the Table

A common metric for evaluating an adversary's uncertainty in finding out the link between a user's old and new pseudonym in a mix zone is calculating its entropy. Consider a sequence of entering/exiting nodes traversing a mix zone i over a period of T time steps, the entropy is given by:

$$H_T(i) = -\sum_i (P_i.log_2 P_i) \tag{1}$$

where P_i are probabilities of possible outcomes. The higher they are, the more the uncertainty. The conditional probability of $X_i = x_i$ given that $\Pi_i = \pi_i$ is:

$$p(x_i|\pi_i) = \frac{p(\pi_i|x_i).P(x_i)}{p(\pi_i|x_i).P(x_i) + p(\pi_i|x_i^c).P(x_i^c)} \tag{2}$$

where $P(x_i)$ is the probability of event x_i occurring; $p(\pi_i|x_i^c)$ is the conditional probabilities of event π_i given that event $P(x_i)$ has not occurred; and $P(x_i^c)$ is the probability of event $P(x_i)$ not occurring. The condition entropy is:

$$H(X_i|\Pi_i) = -\sum_{x_i,\pi_i} p(x_i,\pi_i) log_2 p(x_i|\pi_i), \tag{3}$$

where $p(x_i, \pi_i)$ is the marginal probability of $X_i = x_i$ and $\Pi_i = \pi_i$. For a total number of m mix zones, the Overall Conditional Entropy (OCE) is given by:

$$OCE = \frac{1}{m}\sum_{i=1}^{m} H(X_i|\Pi_i) \tag{4}$$

It therefore follows that for a sample of k mix zones selected to offer user abstraction, their Sample Conditional Entropy (SCE) is thus given by:

$$SCE = \frac{1}{k}\sum_{i=1}^{k} H(X_i|\Pi_i) \tag{5}$$

When constructing optimized placement graphs, we need to measure quality of competing network structures. We do this by calculating the *Bayesian Metrics* (for quality) that is given by *Bayesian-Dirichlet metric (BD)* [13]:

$$BD(B) = p(B)\prod_{i=1}^{n}\prod_{\pi_i}\frac{\Gamma(m'(\pi_i))}{\Gamma(m'(\pi_i)+m(\pi_i))}\prod_{x_i}\frac{\Gamma(m'(x_i,\pi_i)+m(x_i,\pi_i))}{\Gamma(m'(x_i,\pi_i))} \tag{6}$$

and *Minimum description length (MDL) metrics* (for compression) given by *Bayesian information criterion (BIC)* [13]:

$$BIC(B) = \sum_{i=1}^{n}(-H(X_i|\Pi_i)N - 2^{|\Pi_i|}\frac{log_2(N)}{2}) \tag{7}$$

4.2 Abstraction Graph - AG

We propose AG (Algorithm 1) that is generated off-line. Assume that we have n vertices (i.e v = 1, 2 ....., n). Sort vertices v ∈ V in ascending order based on values of *mixcost c* (steps 7 - 12). Starting with the lowest value v_1 , check if there exist a pairwise connection between this vertex and the next closest neighbor in the list and if there is, place a mix zone. Repeat the above until all vertices are covered (steps 16 - 20). To capture highest entropies, we select the segment with maximum traffic capacity and calculate entropy (steps 21 - 25). Let Q represent user's feeling safe value which is associated with conditional entropy. Let K represent minimum acceptable level of service availability that is estimated over the previously recorded service availability requests. If $Q < H(X_i|\Pi_i)$, then we say that the user is safe, otherwise not. We group users according to similarity in their; locations, feeling safe(Q) and service availability(K)-(step 26). We select a sample of mix zones satisfying minimum service availability condition K and where $(Q < H(X_i|\Pi_i))$ (steps 28 - 32). We store $Z_{(AG)}$ (step 34).

Algorithm 1. Abstraction Graph *(AG)*

Require: <Graph G ≡ *(V,E,d,c)*>
Ensure: <Abstraction Graph $Z_{(AG)}$>
1: Let $P_{(t)}$:= n be the initial population;
2: let $V^{'}$:= Sorted vertices v ∈ V in ascending order based on values of *mixcost c*;
3: let $S_{(t)}$:= selected population of promising solutions;
4: let $Z_{(AG)}$:= Abstraction Graph;
5: let *mixcost c* := Intersection Mixing Cost at any intersection v;
6: Initialize $V^{'}$, $S_{(t)}$ and $Z_{(AG)}$;
7: **for** all $v \in V$ **do**
8: **for** $i = 1$ to n **do**
9: sort vertices $v \in V$ in ascending order based on values of *mixcost c*;
10: $V^{'}$:= Sorted Vertices
11: **end for**
12: **end for**
13: let v_1 be the least value;
14: add v_1 to $S_{(t)}$;
15: Starting at v_1
16: **for** $i = 2$ to n **do**
17: **if** there are more Vertices in $V^{'}$ to be covered **then**
18: check and locate a pairwise connection with the next closest neighbor in the sorted list and add this vertex to $S_{(t)}$;
19: **end if**
20: **end for**
21: **for** all $S_{(t)}$ **do**
22: **for** $i = 1$ to n **do**
23: for every vertex, select the segment with maximum traffic capacity;
24: **end for**
25: **end for**
26: group users according to their similarity in locations, feeling safe value of Q and service availability value of K;
27: calculate condition entropies (Formula 3): $H(X_i|\Pi_i) = -\sum_{x_i,\pi_i} p(x_i,\pi_i) log_2 p(x_i|\pi_i)$;
28: **while** $(Q \geq H(X_i|\Pi_i))$ **do**
29: select a sample of mix zones from $S_{(t)}$ that satisfy minimum service availability condition K
30: $Z_{(AG)} = Z_{(AG)} + 1$
31: until $(Q < H(X_i|\Pi_i))$;
32: **end while**
33: construct the resulting Graph $Z_{(AG)}$;
34: return and store Abstraction Graph $Z_{(AG)}$;

4.3 Optimized Decision Graph - ODG

We propose ODG (Algorithm 2) that is generated off-line and uses decomposition, chunking and replacement strategy to achieve an optimal solution from AG. The network B is initialized to an empty network that contains no edges. The decision graph $G_{(ODG)}(i)$ for each variable X_i is initialized to a single-leaf graph, containing only probabilities $p(X_i)$. In each iteration, all operators that can be performed on all decision graphs $G_{(ODG)}(i)$ are examined (steps 9 - 16). The quality of competing network structures is achieved by use of Bayesian metrics (Formula 6) as well as calculating Minimum Description Length using BIC (Formula 7) for compression. To allow for diversity, a replacement strategy is used (steps 17 - 33) where a random subset of candidate solutions is first selected from the original population. The new solution is then compared to each candidate solution in the selected subset and the fitness of the most similar candidate solution determined. The new solution will replace the most similar solution of the subset, or if otherwise, discarded. We store B and G_{ODG} (step 36).

Algorithm 2. Optimized Decision Graph *(ODG)*

Require: <Abstraction Graph $Z_{(AG)}$>
Ensure: <Bayesian Network B, Optimized Decision Graph $G_{(ODG)}$>
1: t := 0;
2: Let $P_{(t)}$:= n be the initial population;
3: let $Z_{(AG)}$:= Abstraction Graph;
4: let $G_{(ODG)}(i)$:= single-node decision graph;
5: let $B_{(t)}$:= empty Bayesian network;
6: let $O_{(t)}$:= Generated Offspring;
7: let *mixcost* c := Intersection Mixing Cost at any intersection v;
8: **while** ($t \leq n$) /greedy algorithm for network construction using decision graphs **do**
9: **for** each variable i **do**
10: **for** each leaf l of $G_{(ODG)}(i)$ **do**
11: add all applicable splits of l into $O_{(t)}$;
12: **for** each pair of leaves l_1 and l_2 of $G_{(ODG)}(i)$ **do**
13: add merge(l_1, l_2) into $O_{(t)}$;
14: **end for**
15: **end for**
16: **end for**
17: **for** each offspring X from $O_{(t)}$ /Restricted Tournament Replacement for diversity maintenance **do**
18: Y := random individual from $P_{(t)}$;
19: CXY := *mixcost*(X,Y);
20: **for** i=2 to n **do**
21: Y' := random individual from $P_{(t)}$;
22: CXY' := *mixcost*(X,Y');
23: **if** (CXY' < CXY) **then**
24: Y := Y';
25: CXY := CXY';
26: **end if**
27: **end for**
28: **end for**
29: **if** (fitness(X) > fitness(Y)) **then**
30: replace Y in $P_{(t)}$ with X (Y is discarded);
31: else
32: discard X;
33: **end if**
34: t := t+1;
35: **end while**
36: return and store B, $G_{(ODG)}$;

5 Security and Privacy Analysis

Our goal is to preserve privacy for users. If security of users is guaranteed, they end up feeling safe. Assume that a malicious adversary knows; 1) the exact location of querying user, 2) continuous query content, 3) locations of some POI, 4) AG and ODG Algorithms. The adversary may carry out 3 types of attacks; 1) *Timing attack*, 2) *Transition attack* and 3) *Continuous Query attacks* [10].

In *Timing attack* [10], the adversary observe the time of entry say t_1 and exit time say t_2 for each user entering and exiting the mix zone. For example, consider a case where the time of entry and exit from a mix zone is represented by a and b respectively. The adversary can calculate the cumulative distribution function of the continuous random variable x (x represent a user) as follows:-

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{8}$$

Again, an adversary can calculate the skewness (which is a measure of symmetry or lack of it) of the distribution function of data set as follow:-

$$Skewness = \frac{\sum_{i=1}^{n}(y_i - \overline{y})^3}{(n-1)s^3} \tag{9}$$

where $\overline{y}$ is the mean, s is the standard deviation, and n is the number of data points. The calculated distribution could either be skewed to the right or left. In such a case, the attacker will use this to eliminate some low probable mappings and narrow down to only the high probable ones. In our proposed algorithms, we use non-rectangular mix zone approach proposed in [11] to confuse the adversary.

In *Transition attack* [10] the adversary will use Markov Chain to estimate the transition probabilities of either turning left, right or going straight for each possible turn in the intersection based on previous observations and use these similarities to infer the target user. For example, let m-step transition probability be the probability of transitioning from state i to state j in m steps. Using total probability theorem we get Chapman Kolmogorov equation [12]:

$$p_{ij}^{m+n} = \sum_{k=1}^{\infty} p_{kj}^{n} p_{ik}^{m} = P^{(m+n)} = P^{(m)} P^{(n)} \tag{10}$$

Increased anonymity strength in ODG drastically reduce chances of this attack.

On the other hand, *Continuous Query (CQ) attacks* [11] are mitigated by increasing the anonymity strengths of the mix-zones [10]. As supported by experiments, the proposed Optimized Decision Graph Algorithm offer greater anonymity strength and therefore minimizing chances of all kinds of *CQ* attacks.

6 Experiments and Evaluations

We adopt the real world mobility trace of human mobility data collected from five different sites from CRAWDAD [6] to generate our trajectory. We rank the

sites according to traffic density starting with the highest densities in New York City, then Disney World - Florida, North Carolina State University (NCSU) Campus, South Korea University (KAIST) Campus and lastly State Fair streets with the lowest density. We form hierarchical decision graph with New York City (mostly with major roads) as our root node and generate ODG.

6.1 Privacy Metrics

1) **Conditional Entropy** - We utilize Formula (1) to calculate individual entropies. In a decision graph, a leaf node will depend on its parent node. we capture dependencies by calculating conditional entropies using Formula (3).
2) **Conditional Probability** (Formula (2)) measures resilience of the proposed algorithms to withstand attacks like Continuous Query (CQ) attacks.
3) **Cost** - Let R be the cost (per user) that results from mixing at a particular vertex with n participants. The average cost (AC) (per user) is given by:-

$$AC = \frac{\sum_{i=1}^{n} R_i}{n} \tag{11}$$

4) **Quality of Service (QoS)** - Let m represent the total number of sampled mix zones and t_1, t_2, ... t_m be the time take to mix per mix zone. The minimum average service availability K that achieves the desired QoS is given by:-

$$K = \frac{\sum_{i=1}^{m} t_i}{m} \tag{12}$$

6.2 Optimization Effectiveness Strategy

For each graph, we select a sample of 10 mix zones with least cost as per Formula (11) and sort them in ascending order to capture hierarchical attribute. We calculate their entropies as shown in Fig. 4. As such, highest entropy of 7.3 is observed in New York City as compared to lowest entropy of 3.3 in State Fair, North Carolina. High entropy is attributed to traffic capacity in road segments

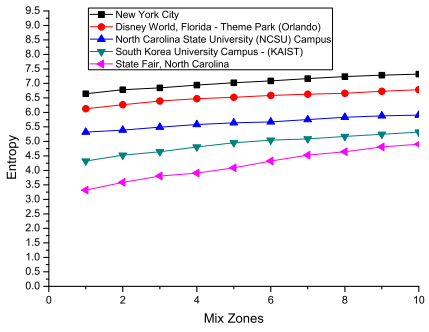


Fig. 4. Entropy Evaluation

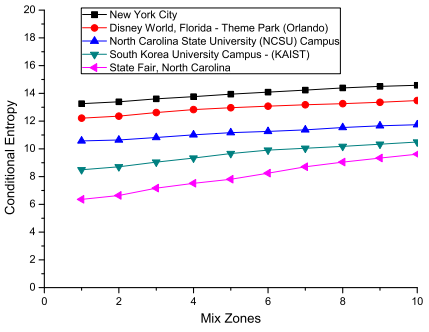


Fig. 5. Conditional Entropy Evaluation

in New York City. More traffic means higher entropy and better privacy. We use Formula (3) to calculate conditional entropies as shown in Fig. 5. We observe that entropy increases to a maximum of 14.6. This is due to dependency factor.

ODG Mix effect is tested in Table 1. The Overall Conditional Entropy (OCE) is calculated using Formula (4). OCE increases to 11.15, achieving greater anonymity.

Table 1. Optimized Decision Graph Mixing effectiveness

Mix Zones	New York	Disney World	NCSU USA	KAIST S Korea	State Fair
1	13.26	12.22	10.57	8.50	6.36
2	13.40	12.36	10.64	8.71	6.64
3	13.60	12.62	10.82	9.05	7.17
4	13.76	12.83	11.01	9.34	7.52
5	13.94	12.97	11.17	9.66	7.81
6	14.09	13.08	11.26	9.91	8.25
7	14.24	13.18	11.37	10.05	8.71
8	14.39	13.26	11.54	10.17	9.05
9	14.51	13.36	11.67	10.34	9.34
10	14.59	13.48	11.74	10.50	9.61
CE	**13.98**	**12.94**	**11.18**	**9.62**	**8.05**
				OCE	**11.15**

6.3 Quality of Service (QoS)

We use Formula (12) to test QoS when K = 0.6 and 1.0. In Fig. 6, we observe that a sample of 8 and 3 mix zones satisfy the minimum set service availability when K = 0.6 and 1.0 respectively. QoS is less when K = 1.0 as opposed to when K = 0.6, indicating a trade-off between privacy and utility (cost and QoS).

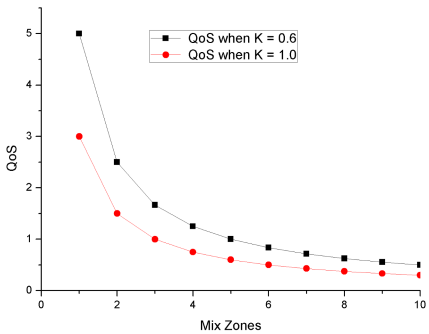


Fig. 6. Quality of Service Evaluation

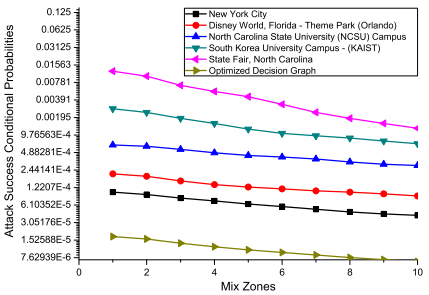


Fig. 7. Attack Resilience Evaluation

6.4 Attack Resilience

From generated ODG, we use Formula (2) to calculate the corresponding conditional probabilities as the user traverse the decision graph from leaf to root nodes and observe that the probability of successful attack is almost zero as shown in Fig. 7. With optimization and proper abstraction, we achieve greater anonymity strength, making it impossible to launch all kinds of C-Q attacks.

7 Conclusion and Future Work

We have presented a mix zone solution of preserving user's privacy using trusted third party architecture. We have proposed AG algorithm that selects a sample of mix zones satisfying user's desired privacy and service availabitity conditions, and ODG algorithm that finds an optimal solution for the placement of mix zones. The results of our experiment show that these Algorithms preserve privacy for users based on their privacy and service availability conditions. We are looking at how to implement decentralized architectures as opposed to centralized ones that are prone to single point of vulnerability as our future work.

References

1. Chow, C.-Y., Mokbel, M.F.: Trajectory privacy in location-based services and data publication. ACM SIGKDD Explorations Newsletter (2011)
2. Chow, C.-Y., Mokbel, M.F., Bao, J., Liu, X.: Query-aware location anonymization for road networks. Geoinformatica (2011)
3. Freudiger, J., Shokri, R., Hubaux, J.-P.: On the optimal placement of mix zones. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 216–234. Springer, Heidelberg (2009)
4. Al-Amin, H., Amina, H., Hye-Kyeom, Y., Jae-Woo, C.: H-star: Hilbert-order based star network expansion cloaking algorithm in road networks. In: 14th IEEE International Conference on Computational Science and Engineering, CSE (2011)
5. Jadliwala, M., Bilogrevic, I., Hubaux, J.-P.: Optimizing mix-zone coverage in pervasive wireless networks. In: JCS (2013)
6. Kotz, D., Henderson, T.: Crawdad, ncsu/mobilitymodels (2009), `http://crawdad.cs.dartmouth.edu/meta.php?name=ncsu/`
7. Xinxin, L., Han, Z., Miao, P., Hao, Y., Xiaolin, L., Yuguang, F.: Traffic-aware multi-mix-zone placement for protec. location priv. In: IEEE INFOCOM (2012)
8. Meyerowitz, J.T., Choudhury, R.R.: Realtime location privacy via mobility prediction: Creating confusion at crossroads. In: Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (2009)
9. Wichian, P., Walisa, R., Nucharee, P.: Navigation without gps: Fake location for mobile phone tracking. In: 11th Intern. Conf. on ITS Telecomm. (2011)
10. Palanisamy, B., Liu, L., Lee, K., Singh, A., Tang, Y.: Location privacy with road network mix-zones. In: IEEE MSN (2012)
11. Palanisamy, B., Liu, L.: MobiMix. Protecting location privacy with mix-zones over road networks. In: IEEE 27th ICDE (2011)
12. Papoulis, A.: Prob., Random Processes and Stoch. Processes. Mc-Graw Hill (2003)
13. Pelikan, M.: Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms. STUDFUZZ, vol. 170. Springer, Heidelberg (2005)
14. Wang, T., Liu, L.: Privacy-aware mobile services over road networks. Proc. of Very Large Databases (VLDB) Endowment 2(1) (2009)
15. Yang, K.-T., Chiu, G.-M., Lyu, H.-J., Huang, D.-J., Teng, W.-C.: Path privacy protection in continuous location-based services over road networks. In: IEEE 8th International Conference on WiMob (2012)

Comparison of Cutoff Strategies for Geometrical Features in Machine Learning-Based Scoring Functions

Shirley W.I. Siu, Thomas K.F. Wong, and Simon Fong

Department of Computer and Information Science
University of Macau
Macau, China
{shirleysiu,mb15404,ccfong}@umac.mo

Abstract. Countings of protein-ligand contacts are popular geometrical features in scoring functions for structure-based drug design. When extracting features, cutoff values are used to define the range of distances within which a protein-ligand atom pair is considered as in contact. But effects of the number of ranges and the choice of cutoff values on the predictive ability of scoring functions are unclear. Here, we compare five cutoff strategies (one-, two-, three-, six-range and soft boundary) with four machine learning methods. Prediction models are constructed using the latest PDBbind v2012 data sets and assessed by correlation coefficients. Our results show that the optimal one-range cutoff value lies between 6 and 8 Å instead of the customary choice of 12 Å. In general, two-range models have improved predictive performance in correlation coefficients by 3-5%, but introducing more cutoff ranges do not always help improving the prediction accuracy.

Keywords: scoring function, protein-ligand binding affinity, geometrical features, machine learning, structure-based drug design.

1 Introduction

With the advances in biophysical experiments in recent years, the amount of known molecular structures has been increased rapidly. Fast and accurate structure-based computational methods to identify putative drug molecules from a large database of small ligand molecules become a crucial step in modern drug discovery [1]. In structure-based drug design (SBDD), the preferred conformation of a ligand molecule in the active site of the target protein is predicted first by a docking algorithm, then the biological activity of the protein-ligand complex in terms of binding constant or binding free energy is estimated using a scoring function [2]. While current docking algorithms are able to generate docked conformations reasonably close to the native complexes, the problem lies in the difficulty to accurately predict the binding affinities of the docked complexes in order to distinguish the active ligands from decoys. In addition, highly

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 336–347, 2013.

accurate scoring functions are essential for lead optimization in the later stage of the drug discovery process.

Despite years of effort, the performance of scoring functions is still far from satisfactory. A recent comparative assessment of scoring functions on a benchmark data set by Cheng et al. [3] has shown that the "ranking power" of the top scoring functions have merely 50% success rate and the correlation coefficients between experimental binding values and predicted scores (the so-called "scoring power") ranged from 0.545 to 0.644 only. When applying an updated list of scoring functions to a new validation data set by Huang et al., the same conclusion was obtained [2]. Both Cheng and Huang's studies highlight the need for new scoring functions with improved prediction accuracy and higher reliability.

One promising alternative to the conventional scoring functions is to apply machine learning (ML) algorithms to construct models for protein-ligand binding prediction. Since ML algorithms learn the theory directly from data through the process of fitting and inference without prior assumption of the statistical distribution in the data, ML is ideally suited for handling biological data that is noisy, complex, and lack of comprehensive theory. Only in the last two years, studies applying ML techniques to construct scoring functions have been seen to emerge. Ballester and Mitchell trained a random forest model (RF-Score) from simple geometrical features of protein-ligand atom-type pairs [5]. Li et al. applied support vector regression (SVR) techniques to learn features derived from knowledge-based pairwise potentials (SVR-KB) and physicochemical properties (SVR-EP) [6]. Another scoring function (NNScore) combined energetic terms from the popular AutoDock Vina program and the BINANA binding characteristics [7] to construct a single-hidden-layer neural network [8]. Recently, an extensive assessment of ML-based scoring functions was carried out by Ashtawy and Mahapatra in which five ML techniques, a combination of three sets of features, and in total 42 different prediction models were constructed [4]. By comparing these ML-based scoring functions to the conventional scoring functions, they showed that the ranking power of the best ML model reaches 62.5% success rate whereas the best conventional model has only 57.8% [4]. Their study is a valuable proof-of-concept that ML is the method of choice for the next generation scoring functions for SBDD. Innovative ML-based scoring functions can be produced by applying new ML algorithms and feature selection methods, or by combining a number of independent ML-models to create consensus scoring functions.

The success of these ML-based scoring functions relies on the correct choice of structural or physiochemical features which can capture the patterns of binding interactions between protein and ligand molecules. In particular, geometrical features such as occurrence count of element pairs are commonly adopted in scoring functions [4,5,6,7]. However, a geometrical feature usually requires a predefined cutoff value to distinguish "interacting" from "non-interacting" atom pairs by distances of the pairs. Often, this value seems to be chosen quite arbitrarily without clear justification. For example, a cutoff of 2.5 Å was used in the BINANA algorithm [7], whereas a cutoff of 12 Å was used by RF-Score [5] and Ashtawys

ML-models adopting the RF-Score features [4]. Kramer and Gedeck defines six cutoff values (3.0, 3.5, 4.0, 4.5, 6.0, 12.0 Å) to bin contact counts by distances of atom-type pairs and ignores all pairs falling out of 12 Å distance. Finer binning strategy is proposed by Hsu et al., where they use 10 cutoff distances (2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0 Å) to count the total number of protein-ligand interactions of vdW force and another 10 cutoff distances (2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4 Å) for hydrogen-bonding and electrostatic interactions [18].

To investigate the predictive abilities of different cutoff strategies in scoring functions, here we compare prediction models using features generated from six cutoff strategies used in literatures, we called them one-range, two-range, three-range, six-range, and soft boundary. Four popular ML techniques – random forests, weighted k-nearest neighbors, support vector machine, and multiple linear regression – are employed to construct in total 24 scoring functions to predict protein-ligand binding affinity. Finally, the best scoring function obtained in this study will be compared to existing conventional and ML-based scoring functions.

2 Materials and Methods

2.1 Data Sets

Data sets used in this study were obtained from the PDBbind database [10]. This manually curated database provides a collection of PDB structures of protein-ligand complexes with experimentally measured binding affinities. Each release of the database contains three data sets: The *general set* is the collection of all binding data in the database; the *refined set* is a selection of high resolution data of 2.5 Å or higher from the general set; the *core set* is a collection of complexes with the highest, medium, and lowest binding affinities from each cluster of protein structures in the refined set. The latest PDBbind v2012 database contains 2,897 complexes in the refined set and 201 complexes in the core set. In this study, the refined set with the core set data removed was used as the training data and the core set was used as the test data.

To compare the ML-based methods to existing scoring functions, we also trained our models using the PDBbind v2007 database, which was used as benchmark data in two recent comparative assessments of scoring functions [3,4].

2.2 Features

Occurrence counts of element pairs were shown to be powerful in predicting protein-ligand binding affinities [5]. Such features are straightforward to compute and are commonly used in combination with other energy or physiochemical features to build prediction models. The determinant to the predictive strength of these geometrical features is the distance criterion used to generate the feature data. For example, an one-range strategy with a single cutoff of 12.0 Å was used in RF-Score [5]; any pair with a distance beyond the cutoff was ignored.

Similarly, a one-range strategy with cutoff at 4.0 Å was adopted in NNScore [8]. A six-range strategy was chosen in Kramer's work; they suggested that different bin sizes should be used for close and distal interactions [17]. It is generally believed that the more the number of distance ranges to use in a model, the higher prediction accuracy would be achieved. However, one additional range introduced in the model will likely to increase the total number of features by double (if no feature selection is conducted), yet the gain in performance may be merely nominal. Therefore, it is instructive to compare the different cutoff strategies and to find out the optimal distance cutoffs which could maximize the predictive performance but minimize the number of required features.

In this work, we are going to investigate one-range, two-range, and three-range cutoff strategies systematically, and compare them to two other cutoff strategies proposed by Kramer [17] and Ouyang [9]. These cutoff strategies are defined as follows:

The one-range strategy counts the total number of a protein element that comes within a cutoff distance (d_0) of a ligand element. Each element-pair is one feature. Using a similar formulation to [5], the occurrence count for a protein-ligand element-pair x_{PL} is:

$$x_{PL} = \sum_{i=1}^{N_P} \sum_{j=1}^{N_L} \theta(d_0 - d_{ij}) \ , \tag{1}$$

where N_P is the number of protein atom which is an element P and N_L is the number of ligand atom which is an element L. d_{ij} is the distance between the i^{th} protein atom and the j^{th} ligand atom of the types in consideration. θ is the unit step function which returns 1 if the argument is positive, and zero if otherwise.

The two- and three-range cutoff strategies introduce more cutoff thresholds such that counts of protein-ligand atoms in different distance ranges are tallied separately. For example, in the two-range cutoff strategy, two cutoffs d_0 and d_1 are defined and so two features x^0_{PL} and x^1_{PL} are created for each element-pair:

$$x^0_{PL} = \sum_{i=1}^{N_P} \sum_{j=1}^{N_L} \theta(d_0 - d_{ij}) \tag{2}$$

and

$$x^1_{PL} = \sum_{i=1}^{N_P} \sum_{j=1}^{N_L} \theta(d_1 - d_{ij}) \times \theta(d_{ij} - d_0) \ . \tag{3}$$

For each of the cutoff strategies, the optimal cutoff value(s) was determined by performing a systematic search over the range of possible values during model construction (see below).

We also tested the six-range cutoff strategy suggested by Kramer [17] and the cutoff strategy with soft boundary from Ouyang [9]. The former uses six thresholds of 3.0, 3.5, 4.0, 4.5, 6.0, and 12.0 Å. For the latter, instead of simply

counting the protein-ligand element pairs, distance-based functions are used to convert each count into two separate contributions, namely repulsion and attraction. Also, instead of a sharp cutoff, a soft threshold for each element pair is determined from their van der Waals radii and a tailing function is introduced at the threshold boundary to account for the reduced intermolecular interaction contribution at large distances.

In this work, nine element types (C, N, O, F, P, S, Cl, Br, and I) are used for both protein and ligand. Therefore, in total there are 81, 162, 243 features for the one-, two-, three-range strategies respectively, 486 features for Kramer's six-range strategy, and 162 features for Ouyang's strategy. Nevertheless, because certain element types (such as F, P, Cl, Br, I) are rare in protein molecules, the maximum number of features for a complex is reduced to 36, 72, 98 for the one-, two-, three-range strategies, and 205, 72 for Kramer's strategy and Ouyang's strategy, respectively.

2.3 Scoring Functions

To compare the predictive performance of different cutoff strategies in scoring functions, we applied four machine learning (ML) techniques popularly used in bioinformatics applications to create prediction models. These ML techniques include random forests (RF) [12], support vector machine (SVM), weighted k-nearest neighbors (wkNN) [13], and multiple linear regression (MLR). All models were trained to predict the pK value.

To compare fairly all prediction models, each model was tuned to have the optimal parameters. The tuning procedure is as follows: For RF, the parameters to be evaluated include the number of trees to grow (*ntree*) and the number of features randomly sampled at each nodal split (*mtry*). Using the training data, ten RF models were generated for each *ntree* value between 500 and 8000 in 500-interval (using the default $mtry = p/3$ where p is the number of features). The optimal *ntree* was determined as the one with the minimum averaged out-of-bag (OOB) error. After deciding the value of *ntree*, similar procedure was conducted to search for the optimal *mtry* value in the range of 1 to the maximum number of features. For SVM models, we used the radial basis function (RBF) as the kernel function. Two parameters – the cost C and the width of the kernel γ – were optimized using grid search. Values which gave the lowest mean squared error in ten-fold cross validation were chosen. For weighted kNN, optimal value for the number of neighbors to consider (k) and the kernel function to convert distances into weights were determined using ten-fold cross validation. The distance metric to use was the Manhattan distance which was found to give better results than Euclidean distance in all cases. Finally, in MLR algorithm, a generalized linear model was fitted to obtain a vector of weights for the input features. Again, ten-fold cross validation was used to evaluate the model.

Training and testing of all prediction models were carried out using the statistical package R [16].

2.4 Performance Metrics

Commonly used metrics to assess the performance of a scoring function are the Pearson's (R_P), Spearman's (R_S) and Kendall's (R_K) correlation coefficients, and root mean squared error ($RMSE$). R_p measures the linear dependence between the predicted binding affinities and the experimental binding affinities:

$$R_p = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y_i} - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y_i} - \bar{\hat{y}})^2}} , \tag{4}$$

where y_i and $\hat{y_i}$ represent the experimental and the predicted binding affinities of sample i. The total number of samples is denoted by N.

R_S is the widely used metric for ranking correlation, i.e. it carries out on the ranks of the values rather than the values themselves. In R_S, it compares the position of a sample when ranked by the predicted binding affinity to its position when ranked by the experimental value:

$$R_s = 1 - \frac{6\sum_{i=1}^{N}(d_i)^2}{N(N^2-1)} , \tag{5}$$

where d_i is the difference in the two ranks for sample i.

R_K is another ranking correlation. It measures the similarity of the two rankings by counting the total number of concordant C (when the order of two samples in the predicted ranking agrees with the order in the experimental ranking) and discordant pairs D (when it disagrees):

$$R_K = \frac{2(C-D)}{N(N-1)} . \tag{6}$$

To measure how well a scoring function predicts the absolute value of the binding affinity, the mean squared error (MSE) or the square root of MSE (RMSE) is used:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2 . \tag{7}$$

3 Results and Discussion

3.1 ML-Models Using Different Cutoff Strategies

One-range cutoff strategy is the simplest strategy where only one cutoff value is used. We built one-range prediction models with cutoff values ranging from 3 to 30 Å and calculated MSE as a function of different cutoffs. As shown in Fig. 1, all models show a change of MSE in a consistent manner: The MSEs are large for models using small cutoff values (typically less than 5 Å). Between 5 and 13 Å, the MSEs reach the minima and vary slightly. Beyond 13 Å, there is a slow but distinctive trend of increase in MSEs. The optimal cutoff for each ML algorithm can be identified as the one with the lowest MSE. Comparison of four

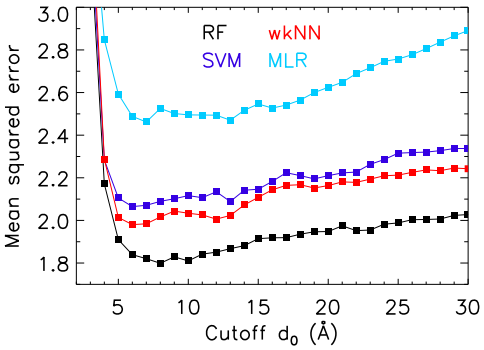


Fig. 1. The mean squared error (MSE) analysis of prediction models using one-range cutoff strategy

ML models on the test data is reported in Table 1. The best scoring function using the one-range cutoff strategy is RF with optimal cutoff at 8 Å. It achieves a correlation coefficient R_P of 0.703, ranking correlations R_S of 0.692 and R_K of 0.504, and RMSE of 1.803.

In the two-range strategy, element-pair counts for short-range interactions are separated from long-range interactions, so two cutoff values d_0 (for the short-range) and d_1 (for the long-range) are required. To find these, we tested prediction models with d_1 between 8 and 14 Å and d_0 between 3 and $(d_1 - 1)$ Å. The result from internal validation is shown in Fig. 2. As expected, all two-range models (color lines) demonstrate improved performances by yielding lower MSEs (except for two cases in MLR) comparing to one-range models (black lines). The optimal cutoff values found in all models are in the ranges of 4-7 Å for d_0 and 11-14 Å for d_1. Again, the RF model achieves the lowest MSE at $d_0 = 7.0$ Å and $d_1 = 11.0$ Å, which attains the correlation coefficient R_P of 0.727, ranking correlations R_S of 0.718 and R_K of 0.527. Compared to one-range models, the predictive performance of two-range models measured by correlation coefficients is increased by 3-5%.

In the three-range strategy, three distance cutoffs are used to separately binning the counts of short-range, mid-range, and long-range interactions. To construct a three-range model, we based on the optimal two-range models to evaluate the addition of a third cutoff threshold in the range of 3 to $(d_1 - 1)$ Å using internal validation procedure. As shown in Table 1, the optimal three-range cutoffs found for the ML-based models are 3-4 Å for d_0, 6-7 Å for d_1, 11-14 Å for d_2. Interestingly, while there is little or no improvement for prediction models using RF, wkNN, and SVM algorithms, MLR has an increase of 5-7% in correlation coefficients compared to its two-range model.

We have also tested a six-range cutoff strategy used by Kramer as part of the descriptor set in their protein-ligand binding scoring function [17]. In this strategy, a smaller bin width is used for counting the short-range interaction and a larger bin width for the long-range interaction. It should be pointed out that

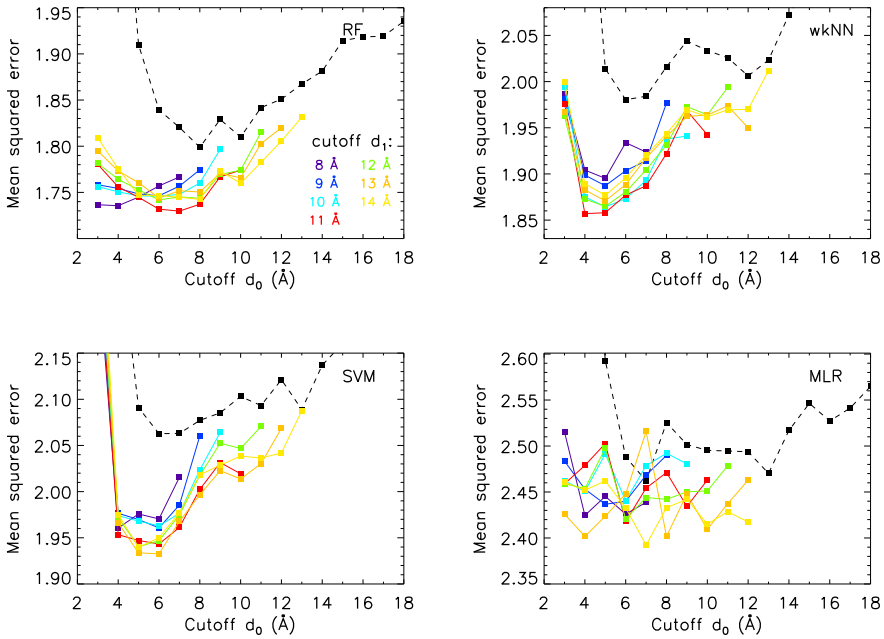


Fig. 2. The mean squared error (MSE) analysis of prediction models using two-range cutoff strategy. Cutoff d_1 varies from 8 to 14 Å and cutoff d_0 from 3 to $d_1 - 1$. Results of one-range cutoff strategy are also shown in black dashed lines for reference.

Kramer assigned atom types to the protein and ligand atoms using Crippen atom typing scheme whereas no typing scheme is applied here. Our purpose is to test if the introduction of more cutoff values would improve the prediction, therefore, all four ML algorithms were applied on six-range features generated using Kramer's cutoff strategy. Except a small improvement of 2% of the wkNN model (compared to its two-range model), in general Kramer's models give worse performance than models using one-range, two-range and three-range cutoff strategies.

The final strategy we tested is the soft threshold method introduced by Ouyang et al. [9]. Unlike the aforementioned strategies where the same cutoff values are used for binning the counts regardless of atom types of the interacting pair, Ouyang introduces a specified cutoff distance for each protein-ligand element pair which is calculated as the sum of their van der Waals radii. This specified cutoff distance defines distance ranges in which the contribution of the pairwise occurrence will be counted in full (a value of 1) or partially (between 0 and 1) as a function of the measured distance. Two values are computed from the so-called membership functions representing the pair's contribution to the repulsive component and the attractive component of the total protein-ligand binding energy. Tailing functions are introduced at the cutoff boundaries such that the repulsive and attractive function at the upper boundary goes smoothly from 1 to 0, and the attractive function at the lower boundary goes from 0 to 1.

Table 1. Performance Comparison of Different Scoring Functions Against the PDBbind v2012 Test Set

Cutoff strategy	ML	Optimal cutoff (Å)	R_P	R_S	R_K	RMSE
One-range	RF	8.0	0.703	0.692	0.504	1.803
	wkNN	6.0	0.691	0.671	0.491	1.778
	SVM	6.0	0.674	0.668	0.479	1.831
	MLR	7.0	0.577	0.587	0.410	2.002
Two-range	RF	7.0, 11.0	0.727	0.718	0.527	1.759
	wkNN	4.0, 11.0	0.682	0.670	0.482	1.796
	SVM	6.0, 13.0	0.676	0.674	0.481	1.802
	MLR	7.0, 14.0	0.582	0.596	0.416	1.990
Three-range	RF	3.0, 7.0, 11.0	0.728	0.720	0.524	1.760
	wkNN	4.0, 7.0, 11.0	0.693	0.681	0.499	1.768
	SVM	4.0, 6.0, 13.0	0.655	0.659	0.474	1.831
	MLR	3.0, 7.0, 14.0	0.611	0.634	0.446	1.942
Kramer	RF	3.0, 3.5, 4.0, 4.5, 6.0, 12.0	0.706	0.700	0.506	1.798
	wkNN		0.690	0.671	0.493	1.769
	SVM		0.652	0.657	0.475	1.860
	MLR		0.556	0.578	0.413	2.013
Ouyang	RF	–	0.717	0.712	0.517	1.771
	wkNN	–	0.669	0.658	0.476	1.811
	SVM	–	0.684	0.688	0.494	1.785
	MLR	–	0.563	0.595	0.419	2.016

Optimal parameters for the ML-based scoring functions are as follows: For RF models, ntree/mtry values are 3000/8 for one-range, 3000/13 for two-range, 4000/15 for three-range, and 3000/55 for Kramer. For wkNN models, triangular kernel is the best kernel in all cases when using Manhattan distance. The optimal k is 11. For SVM models, fine-tuned gamma/cost for radial basis kernel are 0.25/1.414214 for one-range, 0.08838835/2 for two-range, 0.04419417/2.828427 for three-range, 0.015625/2 for Kramer, 0.0625/2 for Ouyang.

To compare how this strategy performs with other cutoff strategies, all four ML algorithms were applied on Ouyang's feature data. The result is also reported in Table 1. Compared to the two-range models which have the same number of features, Ouyang models perform 1-3% worse in correlation coefficients using RF, wkNN, and MLR algorithms, but slightly better using SVM.

Overall, results of the comparative assessment of six different cutoff strategies in this section indicate that the predictive performance of one-range, two-range, and three-range models are within 5% of one another. Introducing more cutoffs with the purpose of attaining finer resolution for contact counts is shown to be unnecessary since the increased model complexity results in poorer predictions. Also, the use of soft boundary to avoid the sharp cutoff do not improve performance significantly. Among models using different ML methods, RF models always outperform other ML models by 1-7%. Therefore, taken into account both the predictive ability and model complexity, we consider the two-range RF model to be the best binding affinity prediction model obtained in this study.

3.2 Comparison with State-of-the-Art Models

To the best of our knowledge, no existing scoring functions have been tested with the new data sets employed in this work. Nevertheless, it is interesting to know how our scoring function is compared to existing conventional and ML-based scoring functions. To this end, two conventional scoring functions, X-Score [14] and DSX [15], were selected for comparison. They are chosen because of their outstanding performances reported in [3]. Programs of these functions are freely available [14] and can be applied directly to predict binding affinities in the test data. As shown in Table 2, the two-range RF outperforms X-Score by 21-25%, 14-29% and 17-35%, in R_P, R_S and R_K, respectively. In recent years, ML-based scoring functions have been showed significant improvements over conventional scoring functions on the previous version of the PDBbind database. Here, we trained our two-range RF scoring function using the PDBbind v2007 refined data set and tested on the PDBbind v2007 core data set. With this, the model can be compared directly to ML-based scoring functions developed with these data sets. These include RF-Score [5], CScore [9], and the five scoring functions from [4] (RF::XA, BRT::AR, SVM::XAR, kNN::XR, MLR::XR, MARS::AR).

Table 2. Performance Comparison of Conventional Scoring Functions Against the PDBbind v2012 Test Set

Scoring function	R_P	R_S	R_K	RMSE
X-Score (HPScore)	0.595	0.625	0.448	1.987
X-Score (HMScore)	0.598	0.631	0.451	1.956
X-Score (HSScore)	0.582	0.604	0.431	2.019
X-Score (AvgScore)	0.600	0.627	0.449	1.969
DSX (PDB)	–	0.594	0.421	–
DSX (Pharm)	–	0.557	0.390	–
Two-range RF (this work)	0.727	0.718	0.527	1.759

Since DSX predicts a score for a protein-ligand complex instead of the binding affinity, R_P and RMSE values of DSX scoring functions were not assessed.

Table 3. Performance Comparison of Existing ML-based Scoring Functions Against the PDBbind v2007 Test Set

Scoring function	R_P	R_S	Ref
CScore	0.801	–	[9]
BRT::AR	0.793	0.782	[4]
Two-range RF	0.787	0.777	this work
RF::XA	0.777	0.771	[4]
RF-Score	0.776	0.762	[5]
SVM::XAR	0.768	0.792	[5]
kNN::XR	0.727	0.720	[5]
MLR::XR	0.641	0.731	[5]
MARS::AR	0.681	0.665	[5]

Parameters for two-range RF: ntree=3000, mtry=13. Results of other scoring functions are taken from literatures.

As shown in Table 3, the performance of the two-range RF model using simple geometrical features is comparable to the top ranked scoring functions using combination of different physiochemical features, and it is ranked as the third best.

4 Conclusion

The development of scoring functions to accurately predict binding affinities of protein-ligand complexes is a daunting task in structure-based drug design. The problem lies in the selection of predictive geometrical or physiochemical features to capture patterns of binding interactions. Among the geometrical features, counting of protein-ligand atomic contacts is a simple yet effective choice as shown in the work of Ballester and Mitchell [5] where only a single cutoff of 12 Å is used to extract contact counts. Numerous cutoff strategies have been employed in other works but used as part of the feature set, so the predictive performance of these cutoff strategies is unclear. In this study, our aim is to compare the predictive abilities of models using different cutoff strategies, namely one-range, two-range, three-range, Kramer's six-range, and the cutoff with soft boundary from Ouyang, and to find out the optimal cutoff values for binding affinity prediction. Prediction models were constructed using four state-of-the-art ML techniques (RF, wkNN, SVM, MLR) with the latest PDBbind v2012 data sets and the models were assessed by three correlation coefficients and RMSD. Our results show that the optimal one-range cutoff value lies between 6 and 8 Å instead of the customary choice of 12 Å . In general, two-range models have improved predictive performance of 3-5% as measured by correlation coefficients, but introducing additional cutoff ranges (three-, six-range) do not always help improving the prediction accuracy. We also show that the two-range RF model (the best model in this work) is able to outperform the best conventional scoring functions and performs comparably to other top-ranked ML-based scoring functions against the PDBbind v2007 benchmark data set. Results of this work are helpful to the selection of geometrical features and cutoff values in the development of scoring functions for structure-based drug design.

Acknowledgments. The authors would like to thank the Information and Communication Technology Office of the University of Macau (UM) for their support of high performance computing facilities. This work is funded by the Research and Development Administration Office of UM (grant SRG022-FST13-SWI).

References

1. Kitchen, D.B., Decornez, H., Furr, J.R., Bajorath, J.: Docking and scoring in virtual screening for drug discovery: Methods and applications. Nat. Rev. 3, 935–949 (2004)
2. Huang, S.Y., Grinter, S.Z., Zou, X.: Scoring functions and their evaluation methods for protein-ligand docking: recent advances and future directions. Phys. Chem. Chem. Phys. 12, 12899–12908 (2010)

3. Cheng, T., Li, X., Li, Y., Liu, Z., Wang, R.: Comparative assessment of scoring functions on a diverse test set. J. Chem. Inf. Model. 49, 1079–1093 (2009)
4. Ashtawy, H.M., Mahapatra, N.R.: A comparative assessment of ranking accuracies of conventional and machine-learning-based scoring functions for protein-ligand binding affinity prediction. IEEE/ACM Trans. Comput. Biol. Bioinf. 9, 1301–1312 (2012)
5. Ballester, P.J., Mitchell, J.B.O.: A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinf. 26, 1169–1175 (2010)
6. Li, L., Wang, B., Meroueh, S.O.: Support vector regression scoring of receptor-ligand complexes for rank-ordering and virtual screening of chemical libraries. J. Chem. Inf. Model. 51, 2132–2138 (2011)
7. Durrant, J.D., McCammon, J.A.: BINANA: A novel algorithm for ligand-binding characterization. J. Mol. Graphics. Modell. 29, 888–893 (2011)
8. Durrant, J.D., Mc Cammon, J.A.: NNScore 2.0: A neural-network receptor-ligand scoring function. J. Chem. Inf. Model. 51, 2897–2903 (2011)
9. Ouyang, X., Handoko, S.D., Kwoh, C.K.: CScore: A simple yet effective scoring function for protein-ligand binding affinity prediction using modified CMAC learning architecture. J. Bioinf. Comput. Biol. 9, 1–14 (2011)
10. Wang, R., Fang, X., Lu, Y., Wang, S.: The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. J. Med. Chem. 47, 2977–2980 (2004)
11. Muegge, I., Martin, Y.C.: A general and fast scoring function for protein-ligand interactions: a simplified potential approach. J. Med. Chem. 42, 791–804 (1999)
12. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
13. Hechenbichler, K., Schliep, K.: Weighted k-nearest-neighbor techniques and ordinal classification. Discussion paper 399, SFB 386 (2004)
14. Wang, R., Lai, L., Wang, S.: Further development and validation of empirical scoring functions for structure-based binding affinity prediction. J. Comput.-Aided Mol. Des. 16, 11–26 (2002), The program X-Score v1.2, `http://sw16.im.med.umich.edu/software/xtool` (August 2013)
15. Neudert, G., Klebe, G.: DSX: a knowledge-based scoring function for the assessment of protein-ligand complexes. J. Chem. Inf. Model. 51, 2731–2745 (2011), The program DSX 0.89, `http://pc1664.pharmazie.uni-marburg.de/drugscore/dsx_download.php` (August 2013)
16. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2012)
17. Kramer, C., Gedeck, P.: Global free energy scoring functions based on distance-dependend atom-type pair descriptors. J. Chem. Inf. Model. 51, 707–720 (2011)
18. Hsu, K.-C., Chen, Y.-F., Yang, J.-M.: GemAffinity: a scoring function for predicting binding affinity and virtual screening. Int. J. Data Mining and Bioinformatics 6, 27–41 (2012)

Bichromatic Reverse Ranking Query in Two Dimensions

Zhao Zhang, Qiangqiang Kang, Cheqing Jin*, and Aoying Zhou

Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, China
{zhzhang,cqjin,ayzhou}@sei.ecnu.edu.cn, kangqiang1107@126.com

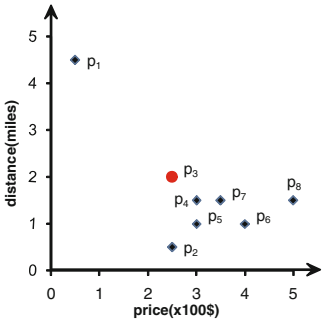
Abstract. Capturing potential customers for a given product based on individual preferences is very important in many personalized applications. Reverse ranking queries are widely employed in this scenario from the perspective of product in database community. Currently, most existing approaches to handle reverse ranking queries generally focus on the d-dimensional space. However, those approaches are oblivious to special properties in the 2-dimensional space which is useful for further optimization. Moreover, there exist many applications, such as data visualization, in the 2-D space. In this work, we propose two general approaches, namely *sorting-based* method and *tree-based pruning* method, in order to efficiently process reverse ranking query in the 2-D space. Both methods are able to handle two variants of reverse ranking query (i.e., *reverse top-k query* and *top-k reverse query*). Analysis and experimental reports on real and synthetic data sets illustrate the efficiency of our proposed methods.

1 Introduction

Ranking query processing is important in the database systems. Typical ranking query can be treated as preference top-k query (top-k query for short), which returns k objects matching user's preference [1] [2]. Usually, a weight vector w is used to represent a user's preference, $\forall i, w[i] \geqslant 0$, and $\sum w[i] = 1$. Top-k queries are popular in many applications. For example, let's consider a hotel case (as shown in Figure 1). Assume a tourist only cares about two factors: the distance to the beach and the price, and her preference weight is described as (0.2,0.8), i.e., she does not care about the price very much. Then, the booking system will recommend three hotels to her by adopting the following scoring function: $f(x, y) = 0.2x + 0.8y$. Hence, three hotels with minimal scores are returned. Clearly, ranking query solves the problem from customers' perspective. However, it cannot answer the query like "who are potential customers of a given hotel" directly. Fortunately, the reverse ranking query is proposed to find potential customers for a given hotel [3]. In the following, we will show a motivated examples about two kinds of reverse ranking queries.

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 348–359, 2013.



User preferences & Rank

user	w[price]	w[distance]	Rank
Alice	0	1	7
Jim	1	0	3
Mary	0.2	0.8	6
Joe	0.7	0.3	3

Fig. 1. An example of reverse ranking query

Example 1. *Figure 1 illustrates a small example about bichromatic reverse ranking containing eight hotels and four customers. First, we compute the rank value of all customer to* p_3*. Consequently, if we want to find who treats* p_3 *as one of her top* 3 *favorite hotels, then Jim and Joe are returned (**reverse top-**k **query**). However, if we want to find the top* 3 *customers who like hotel* p_3 *most, then Jim, Mary and Joe are returned (**top-**k **reverse query**).*

Currently, most existing approaches to process reverse ranking queries are aiming at d-D space, without a special optimization for the 2-D space, such as RTA, the most efficient algorithm [3]. However, it's worth nothing that the 2-D space is important in many applications, such as data visualization. Especially in 2-D space, the weight vectors can be ordered and the hyperplane $\mathcal{H}(w, q)$ will become a straight line $L(w, q)$ [3]. Thus, we can take full advantage of plane geometry's properties to speed up the response time of reverse ranking query on the basis of ordered weights. In the following study, we suppose the users' preferences (i.e., bichromatic) are known and ignore the monochromatic case [4]. Consequently, we propose two general approaches to efficiently process bichromatic reverse ranking query for 2-D space. The two methods process two variants of bichromatic reverse ranking query at the same time, which are *reverse top-k query* and *top-k reverse query*.

We made the following contributions in this paper.

- We propose a general reverse ranking query based on a conditional predicate towards capturing potential customers to a given product. In particular, we can handle two variants of reverse ranking query efficiently at the same time.
- Our proposed sorting-based approach (SA) explores the ordered weight vectors to avoid redundant computations, whereas the tree-based pruning approach (TBA) combines a spatial tree on data points and the ordered weight vectors to reduce unnecessary computations.
- We have conducted extensive experiments over real-life and synthetic data sets to evaluate the efficiency of the proposed methods.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 defines the general reverse ranking query formally. Section 4 and 5 introduce sorting-based approach and tree-based pruning approach. Experimental reports are given in Section 6. Finally, we conclude the paper briefly in the last section.

2 Related Work

Rank-aware query is principle in database community, which has been applied to many applications. Top-k query is the most general rank-aware query, which returns k data points with minimal ranking scores based on a given ranking function. Recently, some researchers proposed another novel query: reverse top-k query, which returns a weight vector set which treat a given point as his/her top-k query result.

Ranking Query. Top-k query retrieves k data points ordered according to a specific ranking function. It has been extensively studied for decades [1] [2] [5] [6] [7] [8]. Fagin's Algorithm (FA) [1] and Threshold Algorithm (TA) [2] are two of famous algorithms. Meanwhile, TA is more efficient than FA. They quickly response top-k queries by using each sorted attributed index. The distributive indexing penalizes query performance because of ignoring attribute correlation.

Another family algorithms of top-k query store pre-processing information off-line, which can help to accelerate response time of online query. Hristidis et al. materializes views of top-k result sets [5]. Chang et al. proposed the Onion indexs [7], and Xin et al. proposed robust index based on Onion index technique [8]. The two index techniques precompute the minimal possible rank of each tuple in the database by considering dominance relationships.

Reverse Ranking Query. Reverse ranking query aims to evaluate the rank of a specified query point in a data points set based on a preference function. The reverse ranking query has many practical applications to determine the importance of a query point among its peers. It is first proposed by Li et.al [9]. And then Xiang Lian et.al. [10] studied reverse ranking queries in uncertain database, Ken C.K.Lee et.al. [11] discussed reverse ranking query over imprecise spatial data. The above three literatures compute the rank of a given point for only one ranking function. We try to simultaneously find the rank of a given point for several ranking functions in most cases.

Our work is most related to reverse top-k query, which is proposed by Akrivi Vlachou et.al. [3] [4] [12]. It returns a weight set S where a given data point belongs to its top-k query result set. However, it can not provide the special optimization for 2-D space. Akrivi Vlachou et.al. proposed to find the most influential data objects using reverse top-k queries [13]. Shen Ge et.al [14] proposed methods for batch evaluation of all top-k queries by the block indexed nested loops and view-based algorithm. It is another approach to answer reverse top-k query by efficiently computing multiple top-k queries. Sean Chester et.al [15] proposed an approach to index reverse top-k queries in two Dimensions.

3 Problem Statement

Let D denote a set containing n points, and each point is described as a 2-dimensional vector $\overrightarrow{p}$. $\forall j \in [1,2]$, $p^{(i)}$ denotes the value of the i-th attribute of $\overrightarrow{p}$, and $p^{(i)} \geqslant 0$. Let W denote a set containing weights of preferences for m customers, and each point is described as a 2-dimensional vector $\overrightarrow{w}$. $\overrightarrow{w}^{(i)}$ denotes the preference weight for the i-th attribute of user $\overrightarrow{p}$, and $\forall i \sum_{i=1}^{2} w^{(i)} = 1 \wedge \wedge w \geqslant 0$.

For any customer with a preference weight w, the utility function $f(\overrightarrow{w}, \overrightarrow{p})$ computes the ranking score of a data point p. The function $f(\overrightarrow{w}, \overrightarrow{q})$ is defined as the inner product of w and p, i.e, $f(\overrightarrow{w}, \overrightarrow{p}) = \sum_{i=1}^{2} w^{(i)} p^{(i)}$. Without loss of generality, we assume that the smaller ranking score values are preferable in this study.

We use rank value to measure relative preference of a customer with preference weight $\overrightarrow{w}$ for data point $\overrightarrow{p}$ compared with other data points in D based on ranking score $f(\overrightarrow{w}, \overrightarrow{p})$ in this study. The rank value is the key in reverse ranking query. Hence, we first define rank value of a customer with preference weight $\overrightarrow{w}$ for a point $\overrightarrow{p}$, and then present the general reverse ranking query.

Definition 1 (rank value, $\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q})$). *Given a point set D, a weight vector $\overrightarrow{w}$, and a query point $\overrightarrow{q}$, the rank of $\overrightarrow{q}$ for $\overrightarrow{w}$ is $\mathsf{rank}(w, q) = |S|$, where $|S|$ is the size of S, a subset of D. For each $\overrightarrow{p} \in S$, we have $f(\overrightarrow{w}, \overrightarrow{p}) < f(\overrightarrow{w}, \overrightarrow{q})$; and for each $\overrightarrow{s} \in D - S$, we have $f(\overrightarrow{w}, \overrightarrow{s}) \geqslant f(\overrightarrow{w}, \overrightarrow{q})$.*

Based on the definition of rank value, we present the formal statement about *general reverse ranking query* as follows.

Definition 2 (bichromatic reverse ranking query, R-rank). *Given a data point set D, a weight set W, R-rank query returns a set S, $S \subseteq W$, such that $\forall \overrightarrow{w} \in S$, $\mathsf{P}(rank(\overrightarrow{w}, \overrightarrow{q}))$, where $\mathcal{P}$ is a conditional predicate.*

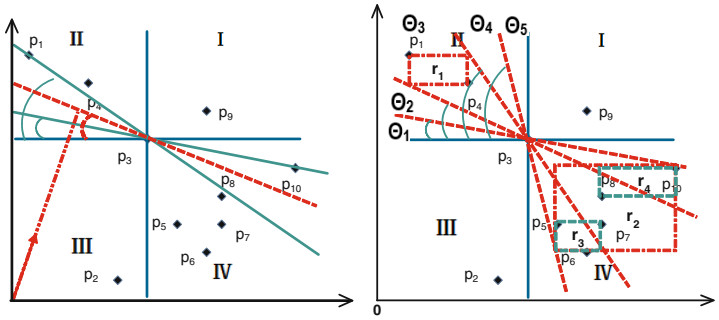
The conditional predicate $\mathcal{P}$ constructs a condition to filter the weight vectors based on $\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q})$. For example, the rank value $\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q}) < k$, top-$k$ $\overrightarrow{w}$ with $min(\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q}))$, and so on.

We give two specific variants of reverse ranking query in terms of the above definition about Bichromatic reverse ranking query.

Definition 3 (bichromatic reverse top-k query). *The* bichromatic reverse ranking query become *bichromatic reverse top-k query when the predicate P is defined by* $\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q}) < k$.

Definition 4 (bichromatic top-k reverse query). *The* bichromatic reverse ranking query become *bichromatic top-k reverse query when the predicate P is defined by top-k $\overrightarrow{w}$ with the smallest $\mathsf{rank}(\overrightarrow{w}, \overrightarrow{q})$.*

For example, in Figure 1, the query point is p_3, *reverse top-k query,* $k = 3$ returns $\{Jim, Joe\}$, (*top-k reverse query*) returns $\{Jim, Mary, Joe\}$.



(a) Sorting based approach (b) Tree based approach

Fig. 2. Sorting approach and Tree based approach

4 Sorting-Based Approach (SA)

The naive approach(NA) needs to evaluate all customer-product pairs. But we find parts of computations can be avoided under the help of sorting of weights. Given a query point $\overrightarrow{q}$, we can divide the 2-D space into four areas, denoted as I, II, III, and IV. Area I contains all points dominated by $\overrightarrow{q}$. Area III contains all points dominating $\overrightarrow{q}$. In the rest two areas, a tuple has no dominance relationship to $\overrightarrow{q}$. Figure 2(a) illustrates an example where $\overrightarrow{p}_3$ is treated as the query point. We only need to consider points in areas II and IV. Let $(\overrightarrow{p}_i, \overrightarrow{q})$ denote a straight line through $\overrightarrow{p}_i$ and $\overrightarrow{q}$, and $h(\overrightarrow{q})$ a horizontal line through $\overrightarrow{q}$. For any point $\overrightarrow{p}$ in II or IV, let $\theta_{\overrightarrow{p}_i}$ denote the angle between $(\overrightarrow{p}_i, \overrightarrow{q})$ and $h(\overrightarrow{q})$.

In 2-dimensional space, given a query point $\overrightarrow{q}$ and a weight vector $\overrightarrow{w}$, there exists one straight line $L(\overrightarrow{w}, \overrightarrow{q})$ which is perpendicular to $\overrightarrow{w}$ and contains $\overrightarrow{q}$. Let $\theta_{\overrightarrow{w}_i}$ denote the angle between $P(\overrightarrow{w}, \overrightarrow{q})$ and $h(\overrightarrow{q})$. We have the following corollary.

Corollary 1. *Given a query point $\overrightarrow{q}$, a point $\overrightarrow{p}$ and a weight $\overrightarrow{w}$. If $\overrightarrow{p}$ is located at area II and $\theta_{\overrightarrow{p}} < \theta_{\overrightarrow{w}}$, we have: $f(\overrightarrow{w}, \overrightarrow{p}) < f(\overrightarrow{w}, \overrightarrow{q})$. If $\overrightarrow{p}$ is located at area IV and $\theta_{\overrightarrow{p}} < \theta_{\overrightarrow{w}}$, we have: $f(\overrightarrow{w}, \overrightarrow{p}) > f(\overrightarrow{w}, \overrightarrow{q})$.*

The correctness of Corollary 1 comes from the semantics of the ranking function. Hence, we can sort all angles for weights and points to compute the query result.

Algorithm 1 is a general solution for answering reverse ranking query. It can not only answer reverse top-k query, but also can answer top-k reverse query. Algorithm 1 describes the steps of our sorting-based approach (SA). The set S contains some entries in a format of $(\theta, o, \mathsf{flag})$, where θ is the angle of two lines discussed above, o is an object (either $\overrightarrow{p}$, or $\overrightarrow{w}$), and flag is a symbol to indicate the source of the object o. If o is a weight vector, flag=weight. Otherwise, flag describes the area which $\overrightarrow{p}$ comes from. At first, the algorithm generates S by scanning W and D with a cost of O(m+n). Second, it sorts all items in S by θ

Algorithm 1. Sorting-based approach (SA) (D, W, $\boldsymbol{q}$, k)

```
1  Let r[w_i] denote the rank of w_i ;
2  Empty a set S;
3  foreach w_i in W do
4      Insert (θ_{w_i}, w_i, weight) into S;
5  foreach p_i in D do
6      if p_i is located at area II or IV then
7          Insert (θ_{p_i}, p_i, area) into S;    //area is II or IV
8  Sort all items in S by θ in ascending order ;
9  count_II ← 0; count_IV = the number of items in IV;
10 foreach triple (θ, o, flag) in A do
11     if flag = weight then
12         r[o] ← count_II + count_IV;
13     else if flag = II then
14         count_II ← count_II + 1;
15     else if flag = IV then
16         count_IV ← count_IV − 1;
17 if the query is reverse top-k query then
18     return r[·] whose value is less than k;
19 if the query is top-k reverse query then
20     return r[·] with the k smallest value;
```

in the ascending order with a cost of $O((m+n)\cdot\log(m+n))$. Finally, it scans all items in S to compute the query result with a cost of $O(m+n)$. As a result, the overall cost is $O((m+n)\cdot\log(m+n))$.

5 Tree-Based Pruning Approach (TBA)

In this section, we discuss answer bichromatic top-k reverse query and bichromatic reverse top-k query based on a R-tree on data points D. Algorithm 2 focuses on answering bichromatic top-k reverse query.

5.1 Answering Bichromatic Top-k Reverse Query

The sorting based approach (SA) needs to visit each $\overrightarrow{p}$ in D. But we find parts of computations can be avoided under the help of an R-tree. As mentioned before, a query point q can divide 2-D space into four areas, and the area II and area IV do not affect the result of reverse ranking query. We only need to consider points in areas II and IV. We build an R-tree to index all data points, and then split the R-tree into two sub-R-trees located in area II and area IV respectively. Let r denote a minimal bounding rectangle (MBR) in R-tree. Let $r.L$ and $r.U$ denote the bottom left and top right points of r respectively. Let $< r.U, q >$

Algorithm 2. Tree-based Pruning Approach (R, W, q, k)

```
1  Let Q and Q' denote two queues;
2  minRank ← the number of objects in R;
3  Split the R-tree into Area II and Area IV based on query point q ;
4  Clear Q and Q'; Set R_b^(w), R_a^(w), R_l and R_u are ∅ ;
5  enqueue(Q, R_II.root);   enqueue(Q, R_IV.root);
6  repeat
7      Copy Q into Q' ;
8      while (r = dequeue(Q') ≠ ∅) do
9          Insert (w_r.U, w_r.L, θ_r.U, θ_r.L) into S;
10     Sort all items S by θ in ascending order, and set θ_0 = 0^o, θ_|S| = 90^o,
       A[i+1] = θ_{i+1} − θ_i ;
11     foreach w in S do
12         foreach r in Q do
13             if (belowL(r, w)) then
14                 R_b^(w) ← R_b^(w) ∩ r ;
15             if (acrossL(r, w_i)) then
16                 R_a^(w) ← R_a^(w) ∪ r ;
17         R_l ← R_l ∩ R_b^(w) ;
18         R_u ← R_u ∪ R_l^(w) ∪ R_a^(w) ;
19     A[i].lb = |R_l|;   A[i].ub = |R_u| ;
20     update minrank by using A[i].ub ;
21     Res ← Res ∪ {A[i]|A[i].ub ⩽ minrank} ;
22     Remove all A[i] in A such that A[i].ub ⩾ minrank or A[i].ub ⩽ minrank ;
23     foreach child node r' of MBR across remaining A[i] do
24         enqueue(Q, r') ;
25 until isQueueEmpty(Q);
26 Compute the exact rank of each weight in all remaining A ;
27 return weights located in Res, and the top k − |{w|w ∈ Res}| weights with
   smallest ranks ;
```

$(< r.L, q >)$ denote a straight line through $r.U$ $(r.L)$ and $\overrightarrow{q}$, and $h(\overrightarrow{q})$ denote a horizontal line through $\overrightarrow{q}$. For any $r.U$ $(r.L)$ in area II or area IV, let θ denote the angle between $< r.U, q >$ $(< r.L, q >)$ and $h(\overrightarrow{q})$.

Corollary 2. *Given a query point q and a point p in area II or IV, the weight vector perpendicular to the straight line $< p, q >$ can be represented by $w = (\frac{k}{k-1}, -\frac{1}{k-1})$, where $k = \frac{p[2]-q[2]}{p[1]-q[1]}$.*

In general, given a query point q and an MBR r of R-tree, the straight line $(< r.L, q >)$ or $(< r.U, q >)$ is corresponding to a unique weight vector $w_{<r.L,q>}$ or $w_{<r.U,q>}$ based on corollary 2. We summarize two kinds of relationships between an MBR r and a straight line l_i below. For example, belowL(r, l_i)=TRUE means

the MBR r is entirely *below* the straight line l_i. In Figure 2(b), the r_1 is below line $< r_1.U, p_3 >$, while r_2 is across line $< r_1.U, p_3 >$.

- belowL(r, l_i). It returns TRUE if and only if $f(w_{l_i}, r.U) < f(w_{l_i}, q)$ always holds.
- acrossL(r, l_i). It returns TRUE if only if $f(w_{l_i}, r.U) < f(w_{l_i}, q)$ and $f(w_{l_i}, r.L) > f(w_{l_i}, q)$ always hold.

We have the following corollary based on the relationships between query point q and the straight line $(r.U)q$ or $(r.L)q$.

Corollary 3. *Given a query point q, and an MBR r in R-tree, we have two angles $\theta_{(r.L)q}$ and $\theta_{(r.U)q}$. If belowL$(r, < r.U, q >)$, then $\theta_{<r.U,q>}.lb = \theta_{<r.U,q>}.lb + |r|$; If acrossL$(r, < r.U, q >)$, then $\theta_{<r.U,q>}.ub = \theta_{<r.U,q>}.ub + |r|$. Similarly, if belowL$(r, < r.L, q >)$, then $\theta_{<r.L,q>}.lb = \theta_{<r.L,q>}.lb + |r|$; If acrossL$(r, < r.L, q >)$, then $\theta_{<r.L,q>}.ub = \theta_{<r.L,q>}.ub + |r|$.*

The correctness of Corollary 3 comes from the semantics of the MBR of R-tree. Hence, we can compute the lower and upper bounds of each angle θ, and can further compute the lower and upper bounds of each angle $\theta_{i+1} - \theta_i$ based on Corollary 3.

Algorithm TBA (Algorithm 2) has two phases. In the first phase, Algorithm 2 traverses R-tree in a BFS (breadth first search) manner. It checks each included angle $A[i]$ to compute the lower and upper rank bounds ($A[i].lb$ and $A[i].ub$)by judging belowL(r, l_i) and acrossL(r, l_i) (at lines 13 and 15). If r is below (or across) $A[i]$, the upper (or lower) rank bound can be updated accordingly. In the second phase, Algorithm 2 prune some angles whose lower bound is larger than *minRank* (at line 22), and return some angles whose upper bound is less than or equal to *minRank* (at line 21). The *minRank* is a global threshold value *minRank*, which ensures at least k weights have a rank value smaller than *minRank*. Formally, consider triples like $(|A[i]|, A[i].lb, A[i].ub)$, where $|A[i]|$ denotes the number of weights in $A[i]$. The value of *minRank* is computed as: $minRank = \text{argmin}_x \left((\sum_{A[i].ub \leqslant x} |A[i]|) \geqslant k \right)$ (at line 20). Based on such information, we have chance to avoid computing the exact rank value for each weight in the angle $A[i]$. But if the angle cannot be pruned and cannot be returned, we append their child nodes to the global queue Q (at lines 23 and 24), and re-execute the first phase of Algorithm 2.

5.2 Answering Bichromatic Reverse Top-k Query

The algorithm for answering reverse top-k query is similar to Algorithm 2. In the first phase, we compute the lower and upper bounds for each angle $A[i]$ by using the same method as in Algorithm 2. In the second phase, there are some differences compared with Algorithm 2. The angles can be pruned whose lower bound is larger than k. And the angles can be returned whose upper bound is less than k. Just as Algorithm 2, we need split the left angles and re-execute operations in the first phase.

Example 2. *Figure 2(b) shows an example of Algorithm 2. Assume query point is* p_3*. We build the angle* $\theta_1 \cdot \theta_5$ *by using min_point and max_point of the top layer MBRs in area II and area IV. And set* $\theta_0 = 0^o$ *and* $\theta_6 = 90^o$*. Let* $A[4]{=}\theta_4 - \theta_3$*, then* r_1 *is below* $A[4]$*, and* r_2 *is across* $A[4]$*. Hence, the* $A[4].lb = 2$*,* $A[4].ub = 7$*. We can get* $A[1].lb = 5, A[1].ub = 5$*;* $A[2].lb = 0, A[2].ub = 5$*;* $A[3].lb = 0, A[3].ub = 7$*; and* $A[5].lb = 2, A[5].ub = 2$ *by the same way. If we answer reverse top-*1 *query, the* $A[1]$*,* $A[4]$ *and* $A[5]$ *can be safely pruned. However,* $A[2]$ *and* $A[3]$ *need to be further probed.*

6 Experiments

In this section, we conduct an extensive experimental evaluation of reverse ranking query. We focus on the evaluation of performance about our novel methods, including SA and TBA. All codes are written in JAVA, and experiments run on the stand-alone computer of Intel CPU/2GHZ and 3GB memory. We present average values over 1000 randomly-chosen queries in all cases.

6.1 Dataset and Setting

We use two kinds of data sets in this study, including a product set D and a customer set W.

Product data set D: We use synthetic and real-life product data sets.

- *Synthetic datasets*: We generate three synthetic datasets, which follow uniform (UN), correlated (CO) and anti-correlated (AC) distribution respectively. See [16] for detailed generation.
- *Real datasets*: We use two real datasets from UCI public datasets, including a bank marketing dataset (45,211 records) [1] and MAGIC gamma telescope dataset (19,020 records) [2].

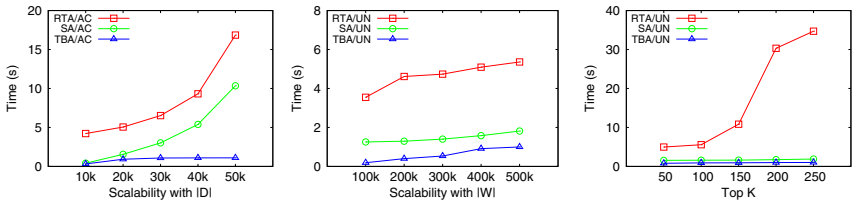
Preference weight data set W: We generate two synthetic data sets, including a uniform set and a clustered set.

- *Uniform distribution*: We repeatedly select one vector from d-dimensional space, and then normalize it to a standard form.
- *Clustered distribution*: This data set is created based on two parameters: g and σ^2. We first randomly select g cluster centroids in d-dimensional space. Then we generate some weights around the centroids with a variance of σ^2 in each dimension. Such method is also adopted by [4].

6.2 Experiment Evaluation

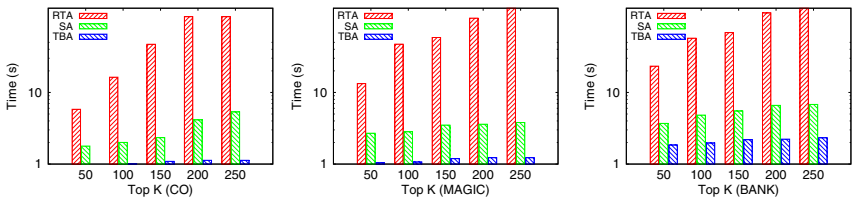
This section studies the performance of our novel algorithms for two kinds of query types. We use time as the metric.

[1] `http://archive.ics.uci.edu/ml/datasets/Bank+Marketing`.
[2] `http://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope`.



(a) Scalability of $|D|$ (b) Scalability of $|W|$ (c) Scalability of k

Fig. 3. Scalability for increasing $|D|$, $|W|$ and k

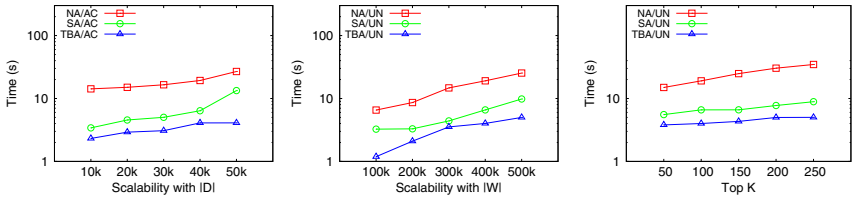


(a) CO data set/CL (b) MAGIC data set/CL (c) BANK data set/CL

Fig. 4. Scalability for synthetic and real data set

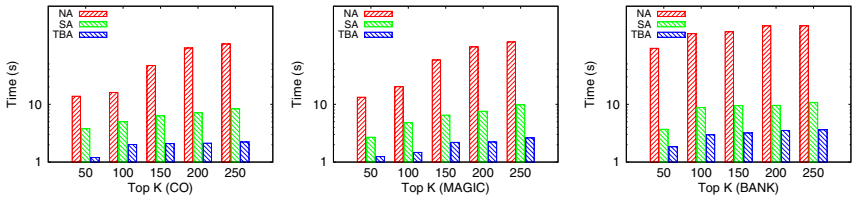
Performance Evaluation of Reverse Top-k Query. In this section, we evaluate the performance of SA and TBA in reverse top-k query for increasing dataset D, weights S and k. We use the RTA as the baseline, which is the most efficient algorithm for computing the reverse top-k RTA [3]. Firstly, we provide a comparison of three algorithms based on the UNIFORM weight set, as shown in Figure 3. Unless mentioned explicitly, We use $|D|$=20k, $|W|$=400k, and k=100. Figure 3(a) shows that when increasing the cardinality of AC data set D, TBA performs better than other two algorithms and RTA has the worst performance. That is because TBA prune computations from perspective of both W and D based on R*-tree and ordered weights. Figure 3(b) shows that with the increment of weight set $|W|$, the performance of all three algorithms slow down. But TBA is still highly efficient. From Figure 3(c), we can obverse that RTA is affected for the increasing k and TBA didn't change a lot. The reason is that the increase of k increases the probability that a query point belongs to top-k for some weighting vector, and therefore the average computations increases, leading to low scalability of RTA.

Thereafter, we study the performance of three algorithms based on the CLUSTER weight set for synthetic data set (CO) and real data set (MAGIC and BANK). We set the cardinality of CLUSTER weight set 100k. The performance of TBA, SA and RTA is in accordance among the cases, and TBA is highly efficient for real and synthetic data set, just as shown in Figure 4.



(a) Scalability of $|D|$ (b) Scalability of $|W|$ (c) Scalability of k

Fig. 5. Scalability for increasing $|D|$, $|W|$ and k



(a) CO data set/CL (b) MAGIC data set/CL (c) BANK data set/CL

Fig. 6. Scalability for synthetic and real data set

Performance Evaluation of Top-k Reverse Query. In this section, we explore the performance of SA and TBA in top-k reverse query, using UNIFORM weight set and CLUSTER weight set respectively, as shown in Figure 5 and Figure 6. We set $|D|$=20k, $|W|$=400k, and top-k=100 by default. Notice that y-axis is in logarithmic scale. First, we compare the performance of TBA with SA and NA (naive algorithm) for increasing dataset D (in Figure 5(a)). TBA performs better than SA and NA, due to the batch pruning strategy. In Figure 5(b), we test the scalability for increasing weight set $|W|$. The chart shows that TBA reduces the response time compared to SA by orders of magnitude. That is because TBA avoids computing each weight in the experiment. In Figure 5(c), we study how the value of k affects the performance of three algorithms. It's clear that TBA is highly efficient for the variation of k. The reason is that it can prune amounts of useless data.

In the sequel, we evaluate the performance of three algorithms for synthetic data set (CO) and real data set (MAGIC and BANK) respectively. We use $|D|$=20k, $|W|$=100k. The weight set conforms clustered distribution. Regarding the real data set, the performance of TBA is in accordance with the case of synthetic data. From Figure 6(a)-6(c), we can observe that TBA behaves more efficiently compared to SA and NA and the benefit increases with k.

7 Conclusion

In this study, we propose two approaches to answer reverse ranking query for 2-D space, including one sorting-based approach (SA) and one tree-based approach (TBA). The two approaches can efficiently process several variants of reverse ranking. Finally, extensive experiment results verify the efficiency of our two approaches upon real and synthetic datasets.

Acknowledgements. Our research is supported by the 973 program of China (No. 2012CB316203), NSFC (60925008, 61070052 and 61370101), Shanghai Knowledge Service Platform Project (No. ZF1213), and Innovation Program of Shanghai Municipal Education Commission(14ZZ045).

References

1. Fagin, R.: Combining fuzzy information from multiple systems. In: PODS, pp. 216–226 (1996)
2. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: PODS, pp. 102–113 (2001)
3. Vlachou, A., Doulkeridis, C., Kotidis, Y., Nørvåg, K.: Reverse top-k queries. In: ICDE, pp. 365–376 (2010)
4. Vlachou, A., Doulkeridis, C., Kotidis, Y., Nørvåg, K.: Monochromatic and bichromatic reverse top-k queries. IEEE Trans. Knowl. Data Eng. 23, 1215–1229 (2011)
5. Hristidis, V., Koudas, N., Papakonstantinou, Y.: Prefer: A system for the efficient execution of multi-parametric ranked queries. In: SIGMOD Conference, pp. 259–270 (2001)
6. Akbarinia, R., Pacitti, E., Valduriez, P.: Best position algorithms for top-k queries. In: VLDB, pp. 495–506 (2007)
7. Chang, Y.C., Bergman, L.D., Castelli, V., Li, C.S., Lo, M.L., Smith, J.R.: The onion technique: Indexing for linear optimization queries. In: SIGMOD Conference, pp. 391–402 (2000)
8. Xin, D., Chen, C., Han, J.: Towards robust indexing for ranked queries. In: VLDB, pp. 235–246 (2006)
9. Li, C.: Enabling data retrieval: by ranking and beyond. PhD thesis, University of Illinois at Urbana-Champaign (2007)
10. Lian, X., Chen, L.: Probabilistic inverse ranking queries in uncertain databases. VLDB J. 20, 107–127 (2011)
11. Lee, K.C.K., Ye, M., Lee, W.C.: Reverse ranking query over imprecise spatial data. COM.Geo, 17:1–17:8 (2010)
12. Vlachou, A., Doulkeridis, C., Nørvåg, K., Kotidis, Y.: Branch-and-bound algorithm for reverse top-k queries. In: SIGMOD Conference, pp. 481–492 (2013)
13. Vlachou, A., Doulkeridis, C., Nørvåg, K., Kotidis, Y.: Identifying the most influential data objects with reverse top-k queries. PVLDB 3, 364–372 (2010)
14. Ge, S., Leong, U., Mamoulis, N., Cheung, D.: Efficient all top-k computation a unified solution for all top-k, reverse top-k and top-m influential queries. IEEE Trans. Knowl. Data Eng. 25, 1015–1027 (2013)
15. Chester, S., Thomo, A., Venkatesh, S., Whitesides, S.: Indexing reverse top-k queries in two dimensions. In: Meng, W., Feng, L., Bressan, S., Winiwarter, W., Song, W. (eds.) DASFAA 2013, Part I. LNCS, vol. 7825, pp. 201–208. Springer, Heidelberg (2013)
16. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: ICDE, pp. 421–430 (2001)

Passive Aggressive Algorithm for Online Portfolio Selection with Piecewise Loss Function

Li Gao[1,2], Weiguo Zhang[1], and Qiang Tang[3]

[1] School of Business Administration, South China University of Technology, Guangzhou, 510640, China
{gaoli,wgzhang}@scut.edu.cn
[2] School of Mathematics, South China University of Technology, Guangzhou, 510640, China
[3] School of Physics Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China
ststq@mail.sysu.edu.cn

Abstract. Passive aggressive algorithms for online portfolio selection are recently shown empirically to achieve state-of-the-art performance in various stock markets. PAMR, one of online portfolio selections, is based on passive aggressive algorithms with an insensitive loss function. Inspired by the mean reversion property and the momentum property of financial markets, we present a passive aggressive algorithm by introducing a piecewise loss function and achieve a novel online portfolio selection strategy named "Passive Aggressive Combined Strategy" (PACS). PACS is able to effectively exploit the power of price reversal and price momentum for online portfolio selection. From our empirical results, we find that PACS can overcome the drawbacks of existing mean reversion algorithms or momentum algorithms and achieve significantly better results. In addition to superior performance, PACS also runs extremely fast and thus is very suitable for real-life large-scale applications.

Keywords: Passive aggressive algorithm, Online learning, Portfolio selection, Momentum, Mean reversion.

1 Introduction

Online portfolio selection (PS), which sequentially selects a portfolio over a set of assets in order to achieve certain targets in the long run, has attracted increasing interests in the fields of machine learning [1-9] and information theory [10-15]. Online PS based on the Capital Growth Theory [16, 17] and information theory focuses on multiple-period or sequential portfolio selection, aiming to maximize portfolio's expected growth rate, or expected log return, which is fitted to the "online" scenario.

Recently, basing on the Capital Growth Theory [16, 17], two approaches for online PS have been proposed. One popular trading idea in reality is *trend following* or *momentum* strategy, which assumes that historically better-performing stocks would still perform better than others in future. Some existing algorithms, such as EG [14] and

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 360–371, 2013.

UP [10], approximate the expected logarithmic daily return and logarithmic cumulative return respectively using historical price relatives. Another widely adopted approach in the learning community is mean reversion [1-4], which is also termed as contrarian approach.

However, financial economists have report that momentum interacts with mean reversion and the combined strategy outperforms both pure momentum strategy and pure contrarian strategy. That is, all existing pure mean reversion strategies or pure momentum strategies by `online algorithms` cannot fully exploit the potential of price fluctuation in stock markets. In this connection, Articles [6-9] proposed the approaches combined "price reversal" and "price momentum" to follow Anticor [1]. Inspired by the philosophy, we introduce a piecewise loss function and present a novel online portfolio selection strategy named "Passive Aggressive Contrarian Strategy" (PACS), which exploits both price momentum and price reversal. By an extensive set of numerical experiments on a variety of up-to-date real datasets, we show that the proposed PACS algorithm significantly surpass PAMR.

The remainder of this paper is organized as follows. Section 2 describes formally defines the problem of online portfolio selection. Section 3 reviews the related work and highlights their limitations. Section 4 presents the methodology employed in this paper. Section 5 discusses data sources and presents the results of extensive empirical studies on real stock markets. This is followed by a conclusion in Section 6.

2 Problem Setting

Let us consider a financial market with m assets, over which we invest our wealth for a sequence of n trading periods. Let us describe the market price change by a sequence of non-negative, non-zero price relative vectors $x_t = (x_t(1), x_t(2), \cdots, x_t(m))$, $x_t \in R_+^m$, $t = 1, 2, \cdots, n$, where R_+^m is the positive orthant. The i^{th} entry $x_t(i)$ of a price relative vector x_t represents the ratio of closing to opening price for the t^{th} trading day.

An investment on the t^{th} period is specified by a portfolio vector $b_t = (b_t(1), b_t(2), \cdots, b_t(m))$ where $b_t(i)$ represents to the portion of the wealth invested in the stock $x_t(j)$ at day t. We also assume the portfolio is self-financed and no margin/short is allowed, therefore each entry of a portfolio is non-negative and adds up to one, that is, $b_t \in \Delta_m$, where $\Delta_m = \{b_t \in R_+^m, \sum_{i=1}^m b_t(i) = 1\}$. The investment procedure is represented by a portfolio strategy, that is, $b_1 = \frac{\mathbf{1}}{m}\mathbf{1}$ and following sequence of mappings. $b_t : R_+^{m(t-1)} \to \Delta_m$, $t = 2, 3, ..., n,$ where $b_t = b_t(x^{t-1})$ is the t^{th} portfolio given past market sequence $x^{t-1} = (x_1, x_2, \cdots, x_{t-1})$ .We denote the portfolio strategy by $b^n = (b_1, b_2, \cdots, b_n)$ for n trading periods.

On the t^{th} period, a portfolio b_t produces a portfolio period return S_t, that is, the wealth increases by a factor of $S_t = b_t^T \cdot x_t = \sum_{i=1}^{m} b_t(i)x_t(i)$. Since we reinvest and adopt price relative, the portfolio wealth would multiplicatively grow. Thus, after n periods, a portfolio strategy b^n produces a portfolio cumulative wealth of S_n, which increases the initial wealth by a factor of $\prod_{t=1}^{n} b_t^T \cdot x_t$, that is

$$S_n(b^n, x^n) = S_0 \prod_{t=1}^{n} b_t^T \cdot x_t$$

where S_0 is set to \$ 1 for convenience.

Finally, we formulate the online portfolio selection problem as a sequential decision problem. The portfolio manager computes the portfolios sequentially and aims to design a strategy b^n to maximize the cumulative wealth S_n. On each period t, the manager has access to all previous sequence of price relative vectors x^{t-1} and previous sequence of portfolio vectors $b^{t-1} = (b_1, b_2, \cdots, b_{t-1})$. On the basis of this historical information, he computes a new portfolio b_t for next price relative vector x_t, where the decision criterion varies among different managers. The portfolio b_t is scored based on portfolio period return S_t. Note that the initial portfolio is set to uniform. The resulting portfolio is evaluated by its portfolio daily return. This procedure is repeated until the end, and the portfolio strategy is finally scored according to portfolio cumulative wealth S_n.

The step of on-line portfolio selection algorithm:

Initialize $S_0 = 1$, $b_1 = (\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m})$;

for each trading day $t = 1, 2, \cdots, n$ do

(1) Portfolio manager learns the portfolio b_t based on historical information;

(2) Market reveals the market price relative x_t;

(3) Portfolio incurs a portfolio daily return $s_t = b^T x_t$.

In the above portfolio selection model, we make assumptions as follows,

1. Transaction cost: we assume no transaction cost or taxes exists in this portfolio selection model;

2. Market liquidity: we assume that one can buy and sell required quantities at last closing price of any given trading period;

3. Impact cost: we assume that market behavior is not affected by a portfolio selection strategy.

3 Related Work

Some common and well-known benchmarks for PS include the Buy-And-Hold (BAH) strategy and the CRP [10, 11]. In our study, we refer to the equal-weighted BAH strategy as the market strategy. Contrary to the static BAH strategy, CRP actively redistributes the wealth among all stocks based on a given portfolio by keeping a fixed fraction at the end of each trading period. The best possible CRP strategy, known as Best CRP (BCRP), is a hindsight strategy. Although nicely grounded in theory, CRP's passive scheme is somewhat limited in achieving good performance.

Some PS strategies assume that historically better-performing stocks would still perform better than others in future. These strategies are also called "momentum strategies" or "follow-the-winner", such as UP and EG, and approximate the expected logarithmic daily return and logarithmic cumulative return respectively using historical price relatives. Cover [10] proposed Universal Portfolio (UP) strategy, where the portfolio is the historical performance weighted average of all CRP experts. Helmbold et al. [14] proposed Exponential Gradient (EG) strategy, which updates the portfolio using multiplicative updates. In essence, EG strategy attempts to maximize the expected logarithmic portfolio period return estimated by last price relative, and minimizes the deviation from last portfolio.

But empirical evidence [18] indicates that such trends may be often violated, especially in the short term. This observation leads to the strategy of buying poor performing stocks and selling those with good performance. This trading principle, known as "mean reversion", is followed by some methods, including Anti-correlation algorithm (Anticor) [1], Online Moving Average Reversion (OLMAR) [2], Confidence Weighted Mean Reversion (CWMR) [3] and Passive Aggressive Mean Reversion (PAMR) [4]. Anticor redistributes the wealth by heuristically exploiting mean reversion via statistical correlations. CWMR strategy exploits the mean reversion property and the variance information of portfolio. OLMAR exploits multi-period mean reversion, which explicitly predicts next price relatives using moving averages, and then learns the portfolio by online learning techniques. Li et al. [4] proposed PAMR, which minimizes the expected return based on last price relatives and only exploits the mean reversion property of financial markets by online passive aggressive learning with a ε -insensitive loss function.

In a word, those approaches mentioned above exploit only price reversal or only price momentum. However, Wu [19] find the combined strategy outperforms both pure momentum and mean reversion strategy. Hence, PAMR cannot fully exploit the potential of price fluctuation in stock markets. Moreover, Articles [6-9] proposed the approaches combined "price reversal" and "price momentum" and achieved better results [1].Therefore, it is particularly critical for online portfolio to exploit not only "price reversal" but also "price momentum" under various stock markets. In the following, we will construct a piecewise loss function for passive aggressive algorithm based on "price reversal" and "price momentum" and then present an online portfolio selection strategy named "Passive Aggressive Combined Strategy" (PACS).

4 Passive Aggressive Portfolio Strategy with Piecewise Loss Function

This section presents the PACS algorithm to capture the dynamics of the momentum and reversals of stock prices in the ultra-short-term or the short-term.

When all stocks drop synchronously or certain stocks drop significantly in the financial crisis, actively rebalancing may not be appropriate since it puts too much wealth on "losers" during the recent financial crisis. To avoid the potential risk concerning such "poor" stocks, PAMR stick to the previous portfolio. As we all known, identifying these "poor" stocks a priori is almost impossible, which is usually known in hindsight. For this reason, PAMR alternates the strategy between "aggressive" CRP and "passive" reversion depending on the market conditions and exploits the mean reversion property [4]. Given a portfolio vector b_t and a price relative vector x_t, Li et al. introduce a ε-insensitive loss function for the t^{th} period as

$$l_{1,\varepsilon}(b;x_t)=\begin{cases}0 & b\cdot x_t \le \varepsilon \\ b\cdot x_t-\varepsilon & \text{otherwise}\end{cases} \tag{1}$$

where $\varepsilon \ge 0$ is the sensitivity parameter which controls the mean reversion threshold. The ε-insensitive loss is zero when return is less than the reversion threshold ε, and otherwise grows linearly with respect to the daily return.

PAMR [4] only adopts the mean reversion trading idea. However, financial economists have report that momentum interacts with mean reversion and the combined strategy outperforms both pure momentum strategy and pure contrarian strategy. Moreover, Articles [6-9] proposed the approaches combining "price reversal" and "price momentum" and achieve better performance in several stock markets. Hence, we would exploit the reversal property and the momentum property about price in stock markets.

Besides the financial crisis, all stocks probably rise synchronously or certain stocks rise significantly, which are called "bull" markets. A bull market is associated with increasing investor confidence, and increased investing in anticipation of future price increases. Under the situations, actively rebalancing may also not be appropriate since the previous portfolio strategy has achieved encouraging performance during the recent bull markets. Thus, to avoid selling "winners", it is a good choice to stick to the previous portfolio, which adopts the momentum property. It is consistent with the idea in the financial crisis.

Let us now describe the basic idea of the proposed strategy. Firstly, if the portfolio daily return is above a certain threshold, we will also try to keep the previous portfolio so that it passively follows the winner to choose the "winner". It is based on price momentum principle that stocks with relatively high returns over the past period should return to investors above average returns over the next period. Secondly, if the portfolio daily return is below a smaller threshold given than before, we will try to keep the previous portfolio such that it passively reverts to the mean to avoid the

potential "loser" stocks. Thirdly, if the portfolio daily return is between two different thresholds, we will actively rebalance the portfolio to ensure that the expected portfolio daily return is below the threshold in the belief that the stock price will revert in the next trading day. According to the phenomenon that the returns exhibit an asymmetric pattern of return reversals, viz., on average, a negative return reverts more quickly, with a greater magnitude, to a positive return than a positive return reverting to a negative one, the stock price will reverts in different degrees if the portfolio daily return falls in different intervals. Since typically portfolio daily return fluctuates around 1, we choose the distance between portfolio daily return and 1 to identify different scenarios. Therefore, we introduce a piecewise loss function to identify the stock market by the following function

$$l_{2,\varepsilon}(b;x_t)=\begin{cases}0 & \left|b\cdot x_t-1\right|\geq\dfrac{\varepsilon}{2}\\ b\cdot x_t-\varepsilon & -\dfrac{\varepsilon}{2}<b\cdot x_t-1\leq 0\\ b\cdot x_t-\dfrac{\varepsilon}{3} & 0<b\cdot x_t-1\leq\dfrac{\varepsilon}{3}\\ b\cdot x_t-\dfrac{\varepsilon}{6} & \dfrac{\varepsilon}{3}<b\cdot x_t-1<\dfrac{\varepsilon}{2}\end{cases} \tag{2}$$

where $\varepsilon\geq 0$ is the sensitivity parameter which controls the mean reversion threshold. The loss is zero when the distance between portfolio daily return and 1 is larger than the reversion threshold $\frac{\varepsilon}{2}$, and otherwise grows linearly with respect to the daily return.

Recalling that b_t denotes the portfolio vector on the t^{th} trading day, the proposed method for (PACS) is formulated as the constrained optimization below:

Optimization Problem (PACS)

$$\begin{aligned} b_{t+1}&=\arg\min_{b\in\Delta_m}\frac{1}{2}\left\|b-b_t\right\|^2+C\xi\\ s.t.\quad & l_{2,\varepsilon}(b;x_t)\leq\xi\\ & \xi\geq 0\end{aligned} \tag{3}$$

where C is a positive parameter to control the influence of the slack variable term on the objection function. We refer to this parameter as the aggressiveness parameter similar to PA Learning [20].

Similarly to the proof in PAMR, we easily derive the approximate solutions for the above PACS formulations using standard techniques from convex analysis [21]. We get the following proposition:

Proposition The solution to the Optimization Problem (PACS) without considering the non-negativity constraint ($b \geq 0$) is expressed as:

$$b = b_t - \tau_t (x_t - \bar{x}_t 1)$$

where $\bar{x}_t = \frac{x_t \cdot 1}{m}$ denotes the market return, and τ_t is computed as: $\tau_t = \min\left\{C, \frac{l_{2,\varepsilon}(b; x_t)}{\left\| x_t - \bar{x}_t 1 \right\|^2}\right\}$.

The main idea of PACS is to devise a loss function in order to exploit the fluctuation of stock prices, that is, if the expected return based on last price relative fall in the interval, the loss will linearly piecewise increase; otherwise, the loss is zero. Based on the loss function, PACS passively maintains last portfolio if the loss is zero, otherwise it aggressively approaches a new portfolio that can force the loss zero.

Input: ε : sensitivity; C : aggressiveness parameter;

Initialize $S_0 = 1$, $b_1 = (\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m})$

Update the portfolio selection:

For $b_1 = 1, 2, \cdots, n$ do

1. Receive stock price relatives: x_t

2. Suffer loss $l_{2,\varepsilon}(b; x_t)$

3. Set step size: $\tau_t = \min\left\{C, \frac{l_{2,\varepsilon}(b; x_t)}{\left\| x_t - \bar{x}_t 1 \right\|^2}\right\}$

4. Update portfolio: $b = b_t - \tau_t (x_t - \bar{x}_t 1)$

5. Normalize portfolio: $b_{t+1} = \arg\min \left\| b - b_t \right\|^2$

Fig. 1. The PACS algorithm

Fig. 1 summarizes the details of the proposed PACS algorithms. The vector b_1 is initialized to uniform portfolio $(\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m})$. At the beginning of t^{th} trading day, we rebalance according to the portfolio determined at the end of last trading day. At the

end of t^{th} trading day, the market reveals a stock price relative vector, which represents the stock price movements. Since both the portfolio and the stock price relatives are already known, we are able to measure the portfolio daily return $b_t^T \cdot x_t$ and the suffering loss $l_{2,\varepsilon}(b; x_t)$ as defined in (2). Then, we calculate an optimal step size τ_t based on last portfolio and stock price relatives. Given the optimal step size τ_t, we can update the portfolio for next trading day. Finally, we perform a normalization step to obtain the final portfolio by projecting the updated portfolio into the simplex domain.

5 Experimental Results

We now evaluate the effectiveness of the proposed PACS algorithm by performing an extensive set of experiments on publicly available and diverse datasets [1-4], which were collected from several diverse financial markets and summarized in Table 1.

The first one is NYSE dataset, which is pioneered by Cover [10] and followed by several other researchers [1-4]. This dataset contains 5651 daily price relatives of 36 stocks in New York Stock Exchange (NYSE) for a 22-year period from Jul.3rd 1962 to Dec. 31st 1984. Li et al. [1] denote this dataset by "NYSE (O)" for short.

The second dataset is the extended version of the above NYSE dataset by Li et al. [1]. They collected the latest data in New York Stock Exchange (NYSE) from Jan. 1st 1985 to Jun.30th 2010, which consists of 6431 trading days. For consistency, Li et al. denote this new dataset as "NYSE (N)".This dataset consists of 23 stocks rather than the previous 36 stocks owing to amalgamations and bankruptcies. All self-collected price relatives are adjusted for splits and dividends, which is consistent with the previous "NYSE (O)" dataset.

Table 1. Summary of4RealDatasets From Stock Markets

Dataset	Market	Time frame	#Days	#Assets
NYSE (O)	Stock(US)	Jul. 3rd 1962-Dec. 31st 1984	5651(day)	36
NYSE (N)	Stock(US)	Jan. 1st 1985-Jun. 30th 2010	6431(day)	23
TSE	Stock(CA)	Jan. 4th 1994-Dec. 31st 1998	1259(day)	88
DJIA	Stock(US)	Jan. 4th 2001-Jan. 4th 2003	507(day)	30

The third dataset "TSE" is collected by Borodin et al. [4], which consists of 88 stocks from Toronto Stock Exchange (TSE) containing price relatives of 1259 trading days from Jan. 4th 1994 to Dec. 31st 1998.

The fourth dataset "DJIA" is collected by Borodin et al. [4]. It consists of Dow Jones 30composite stocks and contains a total of 507 trading days, ranging from Jan. 14th 2001 to Jan. 14th 2003. The statistics in paper [4] show the assumption of mean reversion may not exist in DJIA dataset.

In this work, we follow those above notations. In our experiments, we implement the proposed PACS strategy. We compare it with a number of benchmarks and existing strategies as described in Sect. 3. Below we summarize the list of compared algorithms, whose parameters are set according to the recommendations from their respective studies:

- Market: uniform Buy-And-Hold (BAH) strategy;
- Best-Stock: Best stock in the market in hindsight;
- BCRP: Best Constant Rebalanced Portfolios strategy in hindsight;
- EG: Exponential Gradient (EG) algorithm with the best parameter $\eta = 0.05$ as suggested by Helmbold et al. [14];
- SP: Switching Portfolios with parameter $\gamma = 14$ as suggested by Singer [15];
- ANTI:BAH_{30}(Anticor(Anticor))[1];
- PAMR: Passive Aggressive Mean Reversion Strategy with $\varepsilon = 0.5$ and $C = 500$ as suggested by Bin et al [4].

Table 2reports the main results of this study, that is, the total wealth achieved by various approaches without transaction costs for all four datasets. The top two best results in each datasets are heighted in bold font. The results clearly show that PACS almost achieves the top performance among all competitors except on the DJIA dataset. On the well-known benchmark NYSE(O) dataset, PACS significantly outperforms the state-of-the-art. There is little difference between ANTI and PACS on NYSE (N).Besides, most existing algorithms except ANTI perform badly on the NYSE (N) and DJIA dataset and PACS achieves the best performance on all datasets (except DJIA dataset). Therefore, these results shows that to achieve better investment return, it is more powerful and promising to exploit not only the price reversal property but also the price momentum property for portfolio selection.

Table 2. Total Wealth Achieved By Various Strategies on Four Datasets

Methods	Total Wealth			
	NYSE (O)	NYSE (N)	TSE	DJIA
Market	14.50	18.06	1.61	0.76
Best-stock	54.14	83.5	6.28	1.19
BCRP	250.60	120.32	6.78	**1.24**
EG	27.09	31.00	1.59	0.81
SP	27.08	31.55	1.60	0.81
ANTI	2.31E+08	**6.21 E +06**	39.37	**2.29**
PAMR	**5.14E+15**	1.25E+06	**264.86**	0.68
PACS	**9.16+15**	**1.76 E+06**	**558.47**	0.56

In addition to the final total wealth, we are also interested in examining how the total wealth changes over different trading periods. We plot the wealth curve of total wealth and compare the final wealth of total wealth in Fig.2, where PACS represents the algorithm described in Sec. 4. The x-axis represents the size of the time windows within the coordinate and the y-axis represents the wealth achieved by these algorithms for an initial investment of $ 1.

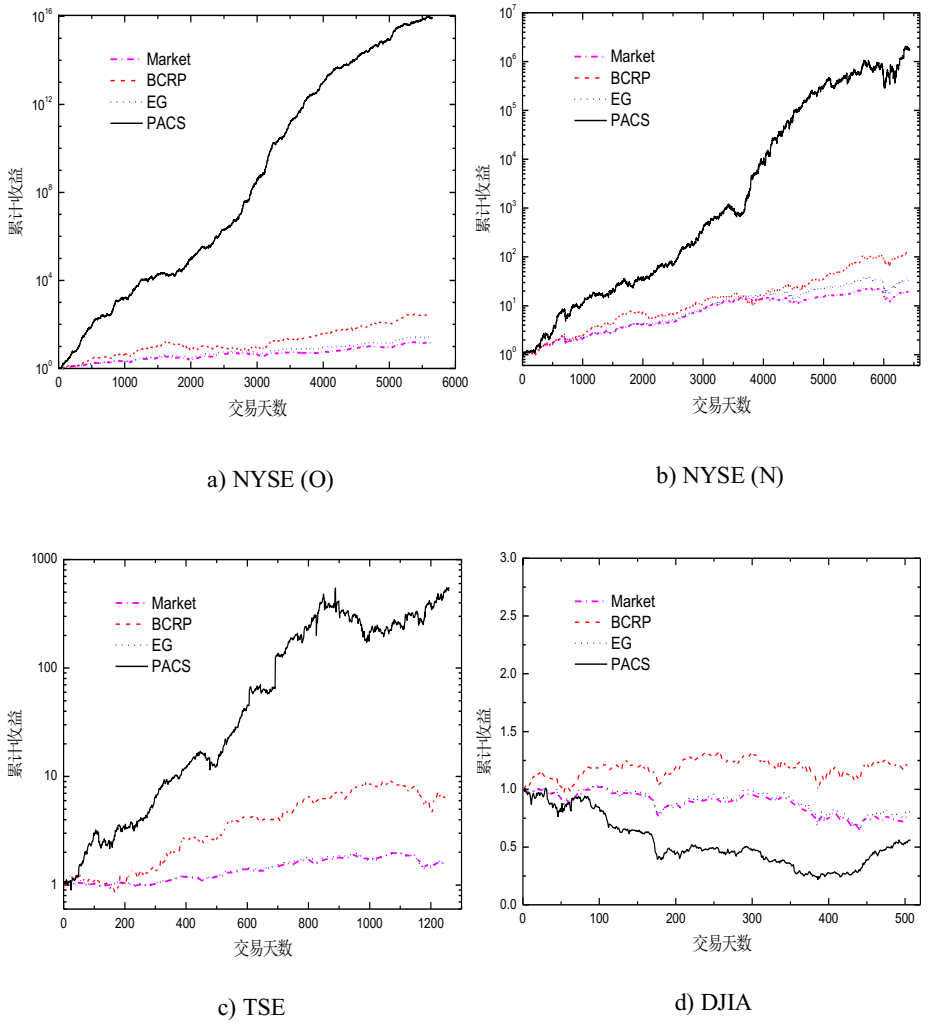


Fig. 2. Trends of total wealth achieved by various strategies during the entire trading periods on the stock datasets

Fig.2 shows the trends of the cumulative wealth by the proposed PACS algorithm and four algorithms (two benchmark sand one state-of-the-art algorithm). From the results, we can see that the proposed PACS strategy consistently surpasses the benchmarks and the competing strategies over the entire trading period on almost datasets (except DJIA dataset).PACS algorithm performs bad on the DJIA dataset, which may be attributed to the reason that the motivating mean reversion does not exist in this dataset. Hence, the results show that the PACS strategy is a promising

and reliable portfolio technique to achieve high confidence. Furthermore, the results again validate the efficacy of the proposed algorithm by a piecewise loss function if appropriately take advantage of price fluctuation in stock markets.

6 Conclusion

This paper constructed a piecewise loss function for passive aggressive algorithm and then proposed an online portfolio selection strategy named "Passive Aggressive Combined Strategy" (PACS) according to the algorithm. Unlike the PAMR algorithm using only the price reversal, the PACS algorithm exploits both the price reversal and the price momentum. PACS successfully applied machine learning techniques for online portfolio selection by capturing the dynamics of the momentum and reversals of stock prices in the ultra-short-term or the short-term. Empirically the PACS algorithm surpasses Market, BCRP, EG, ANTI and PAMR from real markets. However, we find that PACS sometimes fails when the mean reversion property does not exist in the portfolio components. Thus, this proposed strategy can better track the changing stock market, if appropriately used. It also runs extremely fast and is suitable for large-scale real applications.

Acknowledgments. This research was supported by the National Natural Science Foundation of China (No.70825005, 71171086 and 11375278) and Major Project of the National Social Science Foundation of China (No.11&ZD156).

References

1. Borodin, A., El-Yaniv, R., Gogan, V.: Can We Learn to Beat the Best Stock. Journal of Artificial Intelligence Research 21, 579–594 (2004)
2. Li, B., Hoi, S.C.H.: On-Line Portfolio Selection with Moving Average Rever-sion. In: Proceedings of the 29th International Conference on Machine Learning, vol. 1, pp. 273–228. Omnipress, Madison (2012)
3. Li, B., Hoi, S.C.H., Zhao, P., Gopalkrishnan, V.: Confidence weighted mean reversion strategy for on-line portfolio selection. ACM Transactions on Knowledge Discovery from Data 9(4), 1–34 (2012)
4. Li, B., Zhao, P., Hoi, S.C.H., Gopalkrishnan, V.: Passive aggressive mean re-version strategy for portfolio selection. Machine Learning 87(2), 221–258 (2012)
5. Kalai, A., Vempala, S.: Efficient Algorithms for Universal Portfolios. Journal of Machine Learning research 3(3), 423–440 (2003)
6. Gao, L., Zhang, W.G.: Online portfolio selection using ant colony. In: The 2nd IEEE International Conference on Management and Electronic Information, vol. 3, pp. 71–73. IEEE Press, New York (2012)
7. Gao, L., Zhang, W.G.: On-line portfolio selection via mean reversion strategy. Journal of Theoretical and Applied Information Technology 45(1), 136–143 (2012)
8. Gao, L., Zhang, W.G.: Combined mean reversion strategy for on-line portfolio selection. International Journal of Applied Mathematics and Statistics 45(15), 14:349–14:355 (2013)

9. Gao, L., Zhang, W.G.: Weighted moving average passive aggressive algorithm for online portfolio. In: Fifth International Conference on Intelligence Human-Machine Systems and Cybernetics, vol. 1, pp. 327–330. IEEE Press, New York (2013)
10. Cover, T.M.: Universal portfolio. Mathematics Finance 1(1), 1–29 (1991)
11. Cover, T.M.: Universal data compression and portfolio selection. In: Proceedings of the 37th IEEE Symposium on Foundations of Computer Science, pp. 534–538. IEEE Press, Los Alamitos (1996)
12. Iyengar, G.: Universal investment in markets with transaction costs. Mathematical Finance 15(2), 359–371 (2005)
13. Fagiuoli, E., Stella, F., Ventura, A.: Constant rebalanced portfolios and side-information. Quantitative Finance 7(2), 161–173 (2007)
14. Helmbold, D.P., Schapire, R.E., Singer, Y., Warmuth, M.K.: On-line port-folio selection using multiplicative updates. Mathematical Finance 8(4), 325–347 (1998)
15. Singer, Y.: Switching portfolios. International Journal of Neural Systems 8(4), 488–495 (1997)
16. Kelly Jr., J.L.: A new interpretation of information rate. Bell Systems Technical Journal 35(4), 917–926 (1956)
17. Jarrow, R.A., Maksimovic, V., Ziemba, W.T.: Capital Growth Theory. In: Finance Handbook in OR & MS, vol. 9, pp. 123–144 (1995)
18. Jegadeesh, N.: Evidence of predictable behavior of security returns. Journal of Finance 45(3), 881–898 (1990)
19. Wu, Y.R.: Momentum Trading, Mean Reversal and Over reaction in Chinese Stock Market. Review of Quantitative Finance and Accounting 37, 301–323 (2011)
20. Crammer, K., Dekel, O., Keshet, S., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
21. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, New York (2004)

Mining Item Popularity for Recommender Systems

Jilian Zhang[1], Xiaofeng Zhu[1], Xianxian Li[1], and Shichao Zhang[1,2,⋆]

[1] College of CS & IT, Guangxi Normal University, Guilin, 541004, China
[2] The Centre for QCIS, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
{zhangjl,zhuxf,lxx,zhangsc}@mailbox.gxnu.edu.cn

Abstract. Recommender systems can predict individual user's preference (individual rating) on items by examining similar items' popularity or similar users' taste. However, these systems cannot tell item's long-term *popularity*. In this paper, we propose an algorithm for predicting item's long-term popularity through *influential* users, whose opinions or preferences strongly affect that of the other users. Consequently, choices made by certain influential users have the tendency to steer subsequent choices of other users, hence setting the popularity trend of the product. In our algorithm, specifically, through judicious segmentation of the rating stream of an item, we are able to determine whether it is popular, and whether that is the consequence of certain influential users' ratings. Next, by postulating that similar items share similar influential users, and that users rate similar items consistently, we are able to predict the influential users for a new item, and hence the popularity trend of the new item. Finally, we conduct extensive experiments on large movie rating datasets to show the effectiveness of our algorithm.

Keywords: Item Popularity, Recommender System, Collaborative Filtering.

1 Introduction

The easy accessibility of recommendations or comments from experts, friends, colleagues, and famous websites is exerting an increasingly greater influence on consumers' purchasing behavior [3] in the era of web 2.0 [7,10,11]. People would pour through detailed reviews written by professional and casual users on C|net or ZDnet, before deciding on a certain brand or model of electronic camera or laptop. Others surf IMDB.com or Amazon.com for user opinions to determine whether a newly released movie or a new book is worth buying. By studying how certain published opinions are likely to impact the appeal of a product to particular groups of consumers, a retailer could gain valuable insights for formulating his sales strategy. The problem of relating past recommendation to future user preference has been studied extensively in the context of collaborative filtering (CF for short) recommender system, which is an appealing research topic in Web Intelligence. CF methods [1,16] have been proposed to address the problem and some of them have been deployed successfully by commercial websites such as eBay, eLance and Amazon. Specifically, the CF algorithms are used to solve this kind of prediction problem: given a user preference database on a set of

⋆ Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 372–383, 2013.

items, one wants to predict a specific user A's preference (in the form of rating) for an item x. This task can be tackled using (1) item-based methods, which predict A's rating on x by analyzing A's past preferences on other items similar to x, and (2) user-based methods, which estimate the rating result based on preferences on x expressed by other users who have taste similar to that of A. Unfortunately, those methods are designed to predict individual preference that is in the form of a single rating number, resulting in the fact that there has been little effort on modeling long term popularity of items. And the predicting power of influence users, whose opinions or preferences strongly affect that of the other users, has been neglected by existing CF methods.

The goal of this paper goes beyond individual user ratings, to predicting the popularity of an item across several users over a period of time through influential users. We draw inspiration from research on statistical herding, contagion, and information cascades, which provide evidence that the current demand for a product depends on public information about its past demand [13]. Consequently, the choices made by certain influential users have the tendency to steer subsequent choices of other users, hence setting the popularity trend of the product. For example, by identifying influential users who are likely to rate a product favorably, a merchant could target them early in his promotion campaign, and adjust the product pricing according to their reviews.

In this paper, we propose a method for identifying influential users, which are then utilized for predicting popular items based on past user reviews, in the form of historical rating data. Specifically, we firstly identify popular items from historical rating data through judicious segmentation of item rating stream; next statistical test- t test, is performed on changes of item's rating sequence so that we can determine which user is an influential user; and then by postulating that similar items share the same influential users, and that users rate similar items consistently, we are able to predict the influential users for a new item, the ratings that they are likely to assign, and hence the popularity trend of the new item.

2 Preliminaries

2.1 Related Work

The common goal of data mining techniques in recommender systems is to improve the quality of recommendation through text mining or summarization of online review comments. Hu and Liu's work in [6] apply text mining schemes to summarize customer review data on the Web. In a departure from conventional text summarization, their algorithm generates product features only from review sentences that express the opinions of the writers, and it can identify whether each opinion is positive or negative. The outputs at the end are useful for customers to decide whether to purchase a product, and for manufacturers to track and manage customer opinion on their products. A similar work was conducted by Archak et al [2], in which a hybrid technique combining text mining and econometrics is proposed to derive the quantitative impact of consumer textual reviews on products as a linear function with the help of tensor product technique, so that the pricing power of a product can be computed from the consumer review data. There are also some examples of exploiting both textual and numeric review data in order to make a better recommendation. Different from the above methods, our algorithm

takes into account consumer's numeric comment data (i.e., ratings for product), and it can efficiently identify those users who may exert potential influence on the others and predict future popularity of products.

Extensive studies on recommendation systems have focused on collaborative filtering (CF) of user-item rating data in. Existing CF schemes fall into two main categories: model-based and memory-based approaches. For the model-based approach, a model learned from a training dataset is used to estimate the ratings for active users from prior user ratings. Clustering smoothing model [9,14], aspect model [5], and Bayesian network [15] exemplify this line of work. In contrast, memory-based approaches perform calculations on the entire rating dataset in order to find the K most similar users (or items) to the active user (or item) with respect to some similarity measures, then combine the corresponding ratings of these similar users (or items) by using simple weighted sum or regression [12]. Sarwar et al demonstrated that the item-based method greatly outperforms the user-based method in terms of running time, quality of recommendation, and scalability [12].

As discussed above, traditional CF algorithms have achieved tremendous success in recommender systems, with many novel extensions appeared. However, they only focus on how to make a better prediction for an individual rating through the help of various models that they build using machine learning and statistic techniques. As another new extension of the standard CF algorithm, our algorithm goes beyond predicting individual rating for items, to forecasting long-term popularity of item. We propose the concepts of *influential user* and *popular item* in the context of collaborative filtering, and devise efficient algorithm to identify influential user and popular item, which are combined to predict the long-term popularity of new items.

2.2 Problem Definition

While collaborative filtering (CF) has been employed successfully for personalized recommendation in real-world applications, existing CF methods focus only on predicting individual ratings for active users, i.e., individual users' preference for selected items. However, it would be very valuable to go beyond that, to predicting the long-term *popularity* of an item (Note that the definitions of popularity and influential user will be given in the next section). For example, an online bookstore would like to know whether a newly launched book will be popular with its existing customers, in order to devise an appropriate promotion campaign. In real life, products become popular for a variety of reasons. Obvious ones include high quality and consistent performance. There may also be factors that extend beyond any inherent characteristics of the product. In particular, there is evidence that oftentimes the expressed views of a group of customers (or a single customer) have enough clout to steer the preference of subsequent customers. Such customers are called *influential users*. For example, a new book with highly positive reviews (or ratings) by famous critics has a high probability of enjoying brisk sales. We define the problem of predicting an item's popularity as follows

Given a user-item matrix A with rating time for each individual rating and an item I, find a set of popular items x and a group of influential users u from A, and then predict I's popularity based on A, x, and u.

3 The Predicting Model

In this section, we introduce an algorithm for predicting the long-term popularity of an item. The algorithm is centered on the instrumental role that influential users play in shaping the popularity trend of the item. As summarized in Algorithm 1, the algorithm consists of three major stages, including (1) forming clusters of similar terms from the user-item matrix, then identifying the popular items within each cluster (lines 2-5); (2) identifying the influential users who are likely to be responsible for the popular items in each cluster, as well as characterizing the influence of various ratings from those users (lines 6-8); and (3) combining the popularity trends of similar items with the influence exerted by the influential users, into a prediction of the active item's popularity (line 9-16). The following sections elaborate on each stage of the algorithm.

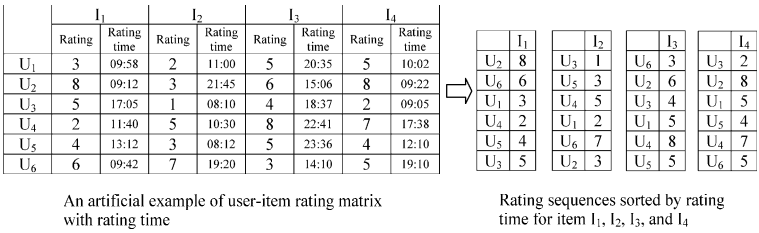
Algorithm 1. Predicting Item Popularity

Input: user-item rating matrix A, an active item X, $\delta, \theta, \lambda, K$
Output: popularity of X

1 Generate rating sequences Q from A; and cluster the items in Q;
2 **for** *each item I in cluster C* **do**
3 Segment item I's rating sequence;
4 Compute the popularity of I;
5 Identify candidate influential users of I;
6 Compute the top-K influential users for each cluster;
7 **for** *each top-K influential user u in cluster C* **do**
8 Compute the influence of u in C under different ratings;
9 **for** *each cluster C* **do**
10 **for** *each item I in C* **do**
11 Compute the similarity between X and I;
12 Compute X's prior popularity $\tilde{H}_C(X|r)$;
13 **for** *each top-K influential user u in C* **do**
14 Predict u's rating for X;
15 Compute X's within-cluster popularity $\hat{H}_C(X)$ in C;
16 Compute X's overall popularity $\hat{H}(X)$;
17 Return $\hat{H}(X)$;

3.1 Identify Popular Items from User Rating Data

The rating sequence R_i of an item i is denoted as $R_i = R_{i1}, R_{i2}, ..., R_{im}$, where m is the number of users. R_{i1} is the rating given by the first user and R_{im} is the score assigned by the latest user. Figure 1 illustrates the transformation of user-item rating matrix to item rating sequences according to the time that user gave rating to items. To find popular items, we apply time series analysis to segment the rating sequences, and look for those that experience sharp spikes in ratings and/or prolonged periods of above-average ratings relative to other similar items.



	I_1		I_2		I_3		I_4	
	Rating	Rating time	Rating	Rating time	Rating	Rating time	Rating	Rating time
U_1	3	09:58	2	11:00	5	20:35	5	10:02
U_2	8	09:12	3	21:45	6	15:06	8	09:22
U_3	5	17:05	1	08:10	4	18:37	2	09:05
U_4	2	11:40	5	10:30	8	22:41	7	17:38
U_5	4	13:12	3	08:12	5	23:36	4	12:10
U_6	6	09:42	7	19:20	3	14:10	5	19:10

An artificial example of user-item rating matrix with rating time

	I_1
U_2	8
U_6	6
U_1	3
U_4	2
U_5	4
U_3	5

	I_2
U_3	1
U_5	3
U_4	5
U_1	2
U_6	7
U_2	3

	I_3
U_6	3
U_2	6
U_3	4
U_1	5
U_4	8
U_5	5

	I_4
U_3	2
U_2	8
U_1	5
U_5	4
U_4	7
U_6	5

Rating sequences sorted by rating time for item I_1, I_2, I_3, and I_4

Fig. 1. An example of how to generate item rating sequence

Segment the Rating Sequence of an Item. The first step in analyzing a rating sequence is to segment it. We adopt the bottom-up Piecewise Linear Representation (PLR) method with slight modification. PLR is a common technique for time series segmentation. PLR variants include sliding windows, top-down, and bottom-up. While the sliding windows has the lowest time complexity and relatively low representation quality, the bottom-up method is able to achieve very good segmentation results with only slightly higher cost [8]. We therefore follow the bottom-up method. In our algorithm, an item's rating sequence is first divided into $m/2$ equal segments, where m is the total number of users who have rated this item. Consecutive segments are then merged iteratively. The merging criterion uses the t-statistic to test whether the difference in mean ratings of two adjacent segments i and j is larger than some threshold δ at some confidence level α. Then the t statistic is compared with the t distribution to determine whether the hypothesis, i.e., the difference in mean ratings is larger than δ, should be rejected [4]. If so, the segments cannot be merged; otherwise segments i and j are merged to form a longer segment. The process is repeated until no adjacent segments can be merged. An example in Figure 2 illustrates how the rating sequence for an item x is segmented.

Compute Item Popularity. Intuitively, the popularity of an item should be judged against other items with similar adoption patterns. For example, it is not useful to compare an educational documentary, which experiences slow adoption over a long life cycle, with the latest mobile phone model that has only a short time span to capture consumers' attention. We therefore cluster the items by the similarity of their rating sequences as advocated in [14].

The similarity measure for the clustering step is the Adjusted Cosine Similarity (ACS), which is widely used for computing item-item similarity. Depending on the intended use of the predicted output, a popular item may be one that receives above average user ratings over prolonged intervals, or substantially higher than average user ratings even if over only a short period. The former suits a merchant who is looking for steady returns, whereas the latter is useful if the merchant is ready to capitalize on demand spikes. The *popularity* of an item x is thus defined as

$$H_c(x) = \lambda \frac{\sum_{l \in S_x} E_{xl}}{\sum_{i \in C} \sum_{l \in S_i} E_{il}} + (1 - \lambda) \frac{\sum_{l \in S_x} L_{xl}}{\sum_{i \in C} \sum_{l \in S_i} L_{il}} \tag{1}$$

where E_{xl} and L_{xl} denote the mean rating and length of segment l of item x, respectively; S_x is the set of x's segments with mean segment rating greater than the mean of

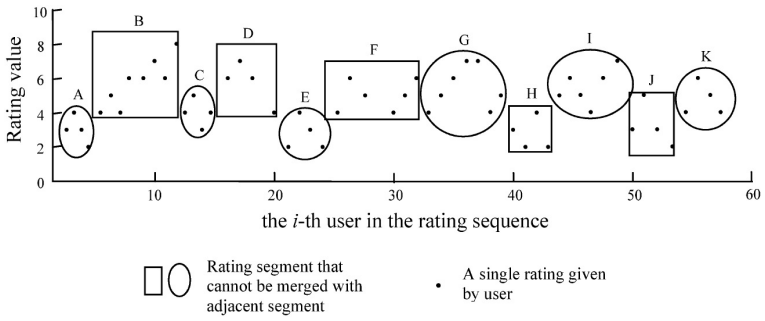


Fig. 2. Segmentation result of an item's rating sequence

the entire rating sequence, i.e., $E_{xl} \geq E_x$ for all $l \in S_x$. $\lambda (0 \leq \lambda \leq 1)$ is a parameter for tuning the relative importance of E_{xl} versus L_{xl}. If $\lambda = 1$, the item popularity relies completely on how high the mean segment ratings of S_x is; if $\lambda = 0$, the item popularity is determined only by the length of the segments in S_x. A higher $H_c(x)$ means that item X is more popular than the others in a same cluster.

3.2 Identify Influential Users Based on Item Popularity

Based on previous research [13], influential users are those whose ratings on an item give a significant boost to the preference of subsequent users. Referring to Figure 2, suppose that D is a high-rating segment. A high-rating segment is one that has a higher mean rating than the item's overall average rating. If the mean of the high-rating segments after D is significantly greater than that of the high-rating segments before D, then the users who contributed to segment D are candidate influential users who potentially are responsible for the improved ratings. The influence levels extended by various rating assignments from the K most influential users are then characterized.

Find Candidate Influential Users. To find the *C*andidate *INF*luential *U*sers (*CINFU*), we perform the following procedure starting with the first segment of each item. Suppose that, in Figure 2, B, D, F, G, and I are the high-rating segments. In processing D, we use the t-test to check whether the average of F, G, and I is significantly greater than that of B. If the difference is significant, we claim that the users who contributed to D are candidate influential users. The test then proceeds to each of the high-rating segments after D, until H_0 is rejected.

Identify Top-K Influential Users. While the above procedure produces the *CINFUs*, we cannot use them directly for predicting popular items. The reasons are: 1) Some *CINFUs* may have strong influence on many items, especially on many popular items, whereas other *CINFUs* may exert only limited influence on a few items. 2) The *CINFU* set may contain false positives, i.e., general users who have low influence on future user preferences. 3) There may be hundreds or even thousands of *CINFUs* for just one cluster of items; obviously, this will degrade the efficiency of our prediction algorithm.

To identify the genuine influential users, a *CINFU* u in cluster C is defined as

$$Score(u, C) = \sum_{x \in C \wedge R_{u,x} \neq 0} \frac{|\overline{R}_B(x) - \overline{R}_A(x)|}{R_{max}} * H_C(x) \quad (2)$$

where x is an item that has been rated by this *CINFU*; $\overline{R}_B(x)$ and $\overline{R}_A(x)$ are the mean ratings of the high-rating segments before and after this *CINFU* in x's rating sequence; R_{max} is the maximum possible rating; and $H_C(x)$ is the popularity of item x in C as defined in Equation 1. The scoring function recognizes u to be an influential user if u has rated many items, and most of those items receive higher mean ratings after u.

Finally, the top-K influential users U_K^C for each cluster C, are the K *CINFUs* in C who possess the largest scores:

$$U_K^C = \{u | Rank(Score(u, C)) \leq K\} \quad (3)$$

This selection process is intended to weed out the false positives, as well as to improve computational efficiency.

Derive the Influence of Top-K Influential Users. Among the influential users (denoted as *INFU*) in U_K^C, the influence exerted by each may still vary greatly. Even for a given *INFU*, the impact of his rating on an item could be conditioned on the actual rating value. For example, a poor rating from a particular *INFU* could sink the popularity of an item, but a favorable rating from the same *INFU* may not have the opposite effect of raising the item's popularity. The converse may also be true. Hence, we need to characterize the influence for each *INFU* for different ratings.

The following formula quantifies the influence of *INFU* on the popularity of an item in cluster C, conditioned on the rating value:

$$INF(u, r) = \frac{\sum_{j \in C} \{H_C(j) * I(R_{u,j}, r)\}}{\sum_{j \in C} H_C(j)} \quad (4)$$

where $I(.)$ is an indicator function such that $I(R_{u,j}, r) = 1$ if $R_{u,j} = r$, otherwise 0; $INF(u, r)$ is the *influence* of *INFU* u across the items in cluster C when u gives rating r. $INF(u, r)$ ranges from 0 to 1 by definition.

3.3 Predict the Popularity of a New Item

In collaborative filtering, individual ratings for an active user are usually predicted with a trained model (in the model-based approach) or other memory-based methods. In contrast, in this paper the predicted popularity trend of active item is derived from our proposed framework.

We observe that many products follow certain familiar adoption patterns. For example, electronic gadgets like mobile phones tend to command consumers' attention when they are launched, but the interest falls quickly over time. In contrast, big ticket items like medical equipments and enterprise software take time to gain acceptance. The existence of common adoption patterns imply that the active item's popularity should bear

close semblance to those of similar items. This observation can be harnessed to predict the popularity of the active item.

Given an active item X, we first determine its prior popularity from each cluster, conditioned upon X receiving rating r. Next, we predict how the influential users of each cluster are likely to rate X, and arrive at the popularity of X judging from that cluster. Finally, summing over the most similar clusters gives X's predicted popularity.

Prior Popularity: The *prior popularity* of item X in cluster C, conditioned upon X being rated r, is:

$$\tilde{H}_C(X|r) = \frac{\sum_{j \in V_r} Sim(X,j) * (H_C(j) - \overline{H}_C(V_r))}{\sum_{j \in V_r} |Sim(X,j)|} \tag{5}$$

Here $Sim(X,j)$ is the ACS similarity between item X and j, V_r is the set of items each of which is rated r by any of the *INFUs* in U_K^C; and $\overline{H}_C(V_r)$ is the mean popularity of all the items in V_r .

Within-cluster Popularity: Next, combining the prior popularities for different r ratings, weighted by the corresponding influence of the *INFUs*, gives X's popularity with respect to each cluster. Specifically, the *within-cluster popularity* of item X in C, $\hat{H}_C(X)$, is the product of the *prior popularity* and the predicted ratings from *INFU* (as given by Equation 4):

$$\hat{H}_C(X) = \sum_r \left(\sum_{u \in U_K^C} INF'(u,r) * \tilde{H}_C(X|r) \right) \tag{6}$$

In the above formula, we need to normalize the influence of *INFUs* in conditioned upon X being rated r, so as to ensure that the *influence* of the *INFUs* are weights that add up to 1. Therefore, we have $INF'(u_j,r) = INF(u_j,r) / \sum_{u_i \in U_K^C} INF(u_i,r)$. An example of how to compute the *within-cluster popularity* is given below.

Overall Popularity: With Equation 6, we could sum up $\hat{H}_C(X)$ over all possible clusters C to arrive at the predicted popularity of X. In practice, those clusters that are similar to X are expected to account for most of the influence on X's popularity. To reduce computation cost, we compute the final *popularity* of X only from C_N, the N nearest clusters of X. Specifically, the overall popularity of X over C_N is the sum of each cluster C's $(C \in C_N)$ *within-cluster popularities*, weighted by the similarity between X and C's centroid:

$$\hat{H}(X) = \frac{\sum_{c \in C_N} Sim(X,C) * \hat{H}_C(X)}{\sum_{c \in C_N} |Sim(X,C)|} \tag{7}$$

4 Empirical Evaluation

In this section, we present a set of comprehensive experiments to study the effectiveness of our proposed framework.

Dataset. We use two MovieLens datasets to carry out our experiments: the first consists of 100,000 ratings for 1682 movies by 943 users; the second has about one million

ratings for 3900 movies by 6040 users (http://www.grouplens.org). In the two datasets, each user has rated at least 20 movies, and each user rating (on a scale of 1 to 5) for an item is associated with a rating time.

Each of the two datasets is randomly divided into a training item set and a test (active) item set, with the split being defined by $Ratio = |TestItems|/|TotalItems|$. In testing each active item, we in turn assume knowledge of its first 5, 10, and 20 user ratings (used for identifying similar items), and the corresponding experiment results are denoted as Given5, Given10, and Given20, respectively. Also, we employ the standard k-means clustering algorithm. Due to space limitation, only results obtained with the first dataset are reported below. Results on the other one show a similar trend.

Measure: We use the *Mean Absolute Error* (MAE) metric to measure the prediction accuracy of our proposed algorithm. Our MAE metric is defined as

$$MAE = \frac{1}{|S|} \sum_{X \in S} |\hat{H}(X) - H(X)| \tag{8}$$

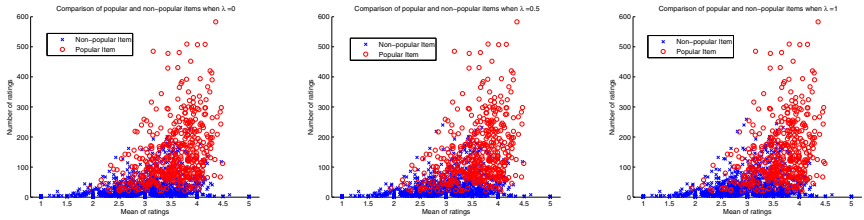
where $\hat{H}(X)$ and $H(X)$ denote the *predicted popularity* and *actual popularity* of active item X in test set S. A smaller MAE in value means a better prediction accuracy.

4.1 Experimental Results Using MAE Metric

Characteristics of Smoothing Factor λ. For the MovieLens 100,000 dataset, if a movie has received many ratings and its mean rating is high, then it is deemed to be popular. For example, some movies have been rated by more than 500 out of 943 users, and have a mean rating of 4 on a scale of 1 to 5. As explained previously, we measure the popularity of an item through the mean rating and length of its segments. δ, the significance level of the difference in mean ratings of successive segments, controls the segmentation of the rating sequence of each item. λ is a parameter for tuning the popularity measure between prolonged above-average ratings and sharp rating spikes.

We begin with several experiments to profile the impact of various λ levels on the popular items identified. We set δ =1, *# of clusters*=3, and arbitrarily fix the threshold at 0.006 (thus items with a popularity that is larger than or equal to 0.006 are identified as popular items). The results are presented in Figure 3, in which the dot and circle represent unpopular item and popular item respectively, while the X and Y axes correspond to the mean rating and number of ratings. The numbers of popular items identified are 455, 441, and 429 out of the 1682 items in the dataset, when λ is set to 0, 0.5, and 1 respectively. Although the numbers of popular items are close across different λ values, we observe that those items with longer high rating segments are more likely to be identified as popular ones with λ=0 in Figure 3(a), whereas items containing segments with high mean ratings are favored with λ=1 in Figure 3(c). When λ is set to 0.5, Figure 3(b) shows that the selection of popular items reflects a tradeoff between the mean rating and the length of the item's high-rating segments.

Characteristics of Top-K INFU. Our procedure in Section 4.2.1 usually generates a very large *CINFU* set, and it is very inefficient to use all the *CINFUs* for predicting the item popularity. So we employ a scoring method to find the top-K *CINFU* in each



(a) $\lambda = 0$ (b) $\lambda = 0.5$ (c) $\lambda = 1$

Fig. 3. Comparison between popular and non-popular items under varied λ

cluster. The underlying assumption is that if a *CINFU* u has rated many items and these items subsequently become popular, then u is likely to be a real *INFU*. The next set of experiments is designed to validate this assumption. The results in Figure 4 are obtained by measuring how many items each of the 943 users has rated (on the X axis)and the mean popularity of these items (on the Y axis), then using Equation 2 and 3 to select the 100 highest-scoring users. The parameter settings are: $ratio$=0.2, *# of clusters*=3, and $Given10$. The circles in the figure represent the top-K *INFUs*. We observe that

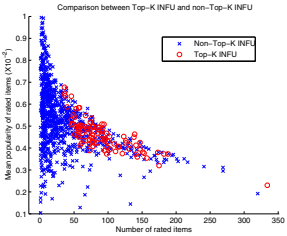


Fig. 4. Top-K INFU vs. non-Top-K INFU

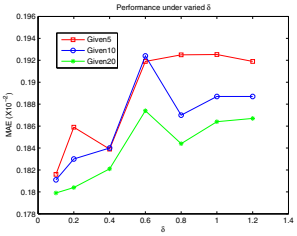


Fig. 5. Performance under different δ

many users have rated several dozens of items, while only a few users have rated more than 100 items. Each of the *INFUs* has rated relatively more items, and these items have larger mean popularities. Selection of the top-K *INFUs* in our algorithm reveals the characteristics of influential users in real life, who generally have higher impacts on certain products and are keen to comment on as many products as they can.

Impacts of δ and θ. In segmenting the rating sequences and in generating the candidate influential users, δ and θ are parameters that set the target difference in mean ratings. Several experiments are performed here in order to show the impacts of δ and θ on the prediction accuracy of our algorithm. The other parameters are fixed at $ratio$=0.2, *# of clusters*=3, λ=0.5, and Top-K users=8. The results are presented in Figure 5 which varies δ with θ=0.4, and Figure 6 which varies θ with δ=0.8.

As illustrated in Figures 5 and 6, $Given5$ results in a larger MAE, whereas $Given20$ gives the smallest MAE for the prediction results. This shows that the algorithm performs better when more user rating information is provided, which is not surprising.

As δ and θ increase, the MAE rises initially, but stabilizes after a while. This happens at $\delta > 0.6$ and $\theta > 0.6$ in Figure 5 and 6 respectively. A large δ value causes most segments with low to moderate mean rating differences to be merged, leaving only adjacent segments that have large gaps between their mean ratings. This adversely affects the identification of popular items, and translates to a fall in prediction accuracy. A similar behavior is observed for θ in Figure 6.

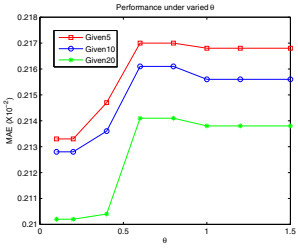


Fig. 6. Performance under different θ

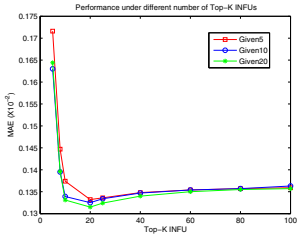


Fig. 7. Various number of Top-K INFUs

Performance under Varied Number of Top-K INFUs. When using the Top-K *INFUs* to predict the active item's popularity, we are confronted with the problem of how many *INFUs* is enough. To study this problem, we run several experiments with $ratio$=0.2, *# of clusters*=3, δ=0.8, θ=0.4, and λ=0.5. The results in Figure 4 show that for the MovieLens 100,000 dataset, the MAE drops rapidly with the initial increase in the number of Top-K *INFUs*. However, after K grows beyond 10, the MAE remains nearly unchanged, meaning that any further increase of *INFUs* does not enhance our algorithm's prediction quality.

5 Conclusions

In this paper, we proposed a novel framework for predicting popular items from historical user-item rating dataset through the help of influential users. We formulate the concepts of popular item and influential users, and quantified them with a method that is built upon the piecewise linear representation algorithm and the t-test. We then harnessed the popularity trends of similar items and predicted ratings of influential users, to predict the popularity of the target item. We have conducted extensive experiments to test the effectiveness of our framework. As an interesting enrichment for recommender systems, our framework is useful in real applications, such as web-based marketing, advertising, and personalized recommendation.

Acknowledgment. This work is supported in part by the Australian Research Council (ARC) under large grant DP0985456; the China "1000-Plan" National Distinguished Professorship; the China 863 Program under grant 2012AA011005; the China 973 Program under grants 2013CB329404, 2012CB326403; the Natural Science Foundation of China under grants 61170131, 61363009, 61263035; the Guangxi Natural Science Foundation under grant 2012GXNSFGA060004; and the Guangxi "Bagui" Teams for Innovation and Research.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Archak, N., Ghose, A., Ipeirotis, P.G.: Show me the money! deriving the pricing power of product features by mining consumer review. In: Proceedings of the 13th ACM SIGKDD, pp. 56–65. ACM (August 2007)
3. Dellarocas, C., Awad, N.F., Zhang, X.: Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. Management Science, 1407–1424 (2003)
4. Doane, D.P., Seward, L.E.: Applied Statistics in Bussiness and Economics. McGraw-Hill, USA (2007)
5. Hofmann, T.: Laten semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS) 22(1), 89–115 (2004)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD, pp. 168–177. ACM (2004)
7. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender systems: an introduction. Cambridge University Press (2010)
8. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 289–296. IEEE (2001)
9. Kohrs, A., Merialdo, B.: Clustering for collaborative filtering applications. In: Proceedings of the CIMCA (January 1999)
10. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the fourth ACM International Conference on Web Search and Data Mining, pp. 287–296. ACM (2011)
11. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proc. of the 5th ACM Conference on Recommender Systems, pp. 157–164. ACM (2011)
12. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of Int. Conf. of World Wide Web, pp. 285–295. ACM (2001)
13. Vany, A.D.: Hollywood economics: how extreme uncertainty shapes the film industry. Routledge, USA (2004)
14. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th ACM SIGIR 2005, pp. 114–121. ACM (2005)
15. Zhang, Y., Koren, J.: Efficient bayesian hierarchical user modeling for recommendation systems. In: Proceedings of the 30th ACM SIGIR, pp. 47–54. ACM (2007)
16. Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J.R., Zhang, Y.-C.: Solving the apparent diversity-accuracy dilemma of recommender systems. PNAS 107, 4511–4515 (2010)

Exploring an Ichthyoplankton Database from a Freshwater Reservoir in Legal Amazon

Michel de A. Silva, Daniela Queiroz Trevisan, David N. Prata, Elineide E. Marques, Marcelo Lisboa, and Monica Prata

Postgraduate Program in Computational Modeling Systems
Federal University of Tocantins (UFT)
Palmas – TO – Brasil
{michel,danielatrevisan,dnnprata,emarques}@uft.edu.br

Abstract. The purpose of this study is to use data mining techniques for the exploratory analysis of a database of ichthyoplankton samples from a freshwater reservoir in Legal Amazon. This database has already been analyzed using statistical techniques, but these did not find a relationship between biotic and abiotic factors. The application of the Apriori algorithm allows us to generate association rules that yield an understanding of the process of fish spawning in Tocantins River. In this case, we demonstrate the effective use of data mining for the discovery of patterns and processes in ecological systems, and suggest that statistical methods often used by ecologists can be coupled with data mining techniques to generate hypotheses.

Keywords: data mining, Apriori, database, ichthyoplankton.

1 Introduction

In recent decades, the evolution of hardware technologies has dramatically increased the amount of stored data. These data are kept in many structures, such as database systems, webpages, conventional files, spreadsheets, tablets, and smartphones.

Information technology, and specifically data mining, has provided information by applying several artificial intelligence techniques. The manipulation of databases has uncovered new approaches to the use of this information, broadening the discussion of the stored data and raising new study questions.

The increasing world population and technological advances in production processes have given rise to a world "guided by resources". The environmental impact of human exploitation is a topic that demands significant effort from the scientific community. Ecological data mining seeks answers to how we understand and use our natural resources more efficiently.

Ecologists have mainly used statistical methods to analyze the relationship between an observed response and a set of predictive variables in a dataset. This approach to

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 384–395, 2013.

data analysis is more appropriate for hypothesis testing. When prior knowledge is minimal and the assumptions are not clearly developed, exploratory analyses are more appropriate than confirmatory analysis [1]. Data mining techniques are often more powerful, flexible, and efficient than statistical methods for conducting these exploratory studies (e.g., [2-4]).

In this paper, we describe the application of the Apriori algorithm [5] and association rules to search for patterns in an ichthyoplankton ecological database. The development of this process has disclosed information from a database domain that could not be detected through the use of statistical methods alone [6]. The application of this technique allows us to determine rules between certain proprieties, thus producing unexpected information that was previously hidden.

Our main goal is to answer the following question: "Is there any relationship between abiotic factors and the larval stage?"

In the first phase of modeling the problem, we pinpointed the topics for which ecologists are seeking answers, as well as relevant information for the application of each of the properties. At the second, preprocessing stage, we selected the data to be processed, integrating, reducing, and transforming them to add quality and ensure their feasibility. From this point, at the third stage, we started the experimental phase of data mining by applying the Apriori algorithm to the preprocessed data, using specified parameters for the degree of confidence and support. The fourth stage, known as post-processing, concerns the refinement and interpretation of the extracted rules, including an expert assessment to analyze the results.

2 Ichthyoplankton Database

In the context of conservation and natural resource management, databases are likely to become more extensive because of the enlargement of data collection, time series, and details of sampling networks. At present, the monitoring of occurrences that are taking place in natural systems, anthropogenic or not, is one of the major challenges to environmental management in tropical areas. Though desirable, the detection and description of these occurrences, identification of variables to indicate environmental quality, and construction and verification of models for future predictions in complex systems (such as aquatic ecosystems), are in the early stages of development in tropical areas. The relationship between high species diversity and environmental factors allows a wide range of scenarios and interactions that require different techniques to be identified.

In this scenario, information from monitoring fish fauna, both in the early stages of development (eggs and larvae) and in adulthood, associated with hydroelectric projects constitute a dataset that has not yet been explored.

In Tocantins River, as well as in other rivers in Brazil and across the world, fish fauna has been systematically modified by the construction of hydroelectric plants.

Seven projects operating in Tocantins River have dramatically changed the environment of fish and wildlife, and these may be further threatened by the construction of other plants already planned for this watercourse [7].

Nevertheless, monitoring the behavior of fish fauna, identifying key spawning sites, and studying the early stages of development are protection strategies that help to define the limits of conservation areas. These strategies can be incrementally improved and refined using the existing database, but they necessarily require the use of information technology.

Even when data collection is in progress, the application of different analytical techniques can help to evaluate sampling networks and spot the gaps that could be filled with the correct set of information.

Monitoring fish fauna in the area of influence of the Lajeado reservoir, on Tocantins River, produced one of the first systematized information bases for a research group in Legal Amazon, as data were being collected before, during, and after the construction of the reservoir. Changes in the distribution and abundance of fish larvae have been reported using traditional methods of analysis [6]. Accordingly, the application of data mining may enhance this research and identify new matters, methodologies, and results.

3 Data Collection

The studies were conducted at the Lajeado reservoir area (HPP Luís Eduardo Magalhães), shown in Figure 1. Construction of the dam was halted in 2002.

Fish larvae data were collected monthly from April 2010 to March 2012 at 12 points located around the reservoir (downstream and upstream) and the headwater and mouth of its major affluents (Santa Tereza, Santo Antônio, São Valério, Crixás, Areias, Matança, Água Suja, and Santa Luzia rivers) using the methodology suggested by [8]. At the same time, information concerning the air temperature, water conductivity, pH, and water transparency were recorded.

Water temperature data (°C) were obtained using a digital thermometer; the transparency of the water column (m) was measured using a Secchi disk of 0.30 m in diameter, while the hydrogen potential (pH) and electrical conductivity ($\mu S.cm^{-1}$) were obtained from portable digital potentiometers, and the concentration of dissolved oxygen was measured using a portable digital oximeter (YSI, $mg.L^{-1}$).

Samples were taken through horizontal hauls by a conic-cylindrical plankton net, with a 500 micron mesh and length of 1.5 m (conical part of 0.9 m and cylindrical part of 0.6 m), and a mechanical flowmeter coupled to the mouth of the network to obtain the volume of filtered water. Samples were taken at a depth of about 20 cm in the middle and near the left and right margins of the sites, which were more than 20 m wide.

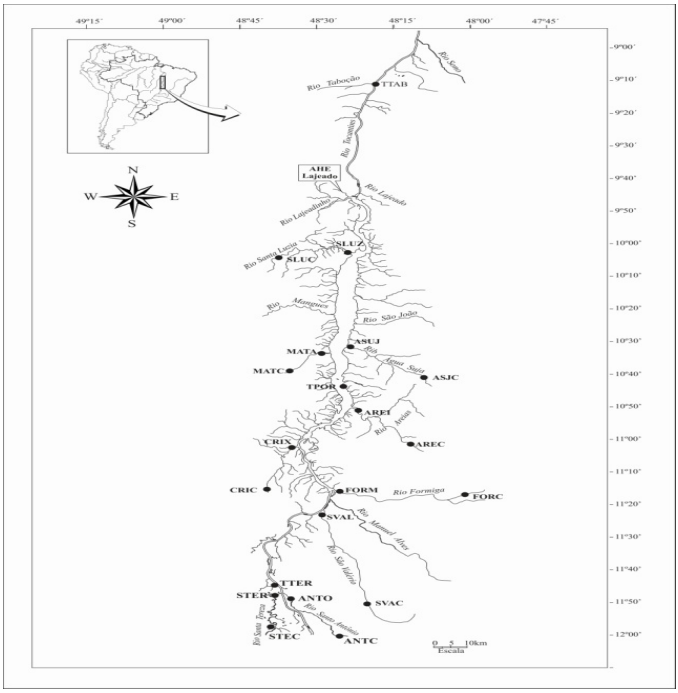


Fig. 1. Location of the sample points in the Lajeado reservoir and its area of influence

The samples were fixed in formalin and analyzed using a stereoscopic microscope. Larvae were separated according to the stage of development (larval yolk, pre-flexion, flexion, and post-flexion) considering criteria such as the presence of a yolk sac, mouth opening, and development of the tail, fin, and supporting elements, following the methodology in [8].

Larvae were identified in terms of their order and family level where possible, according to [8], which considers morphological characters such as the size and position of the eyes, shape and pigmentation of the body, position of the anal opening to the body, forming sequence and position of the fins, presence of wattles, and meristic data, such as the number of myomeres and fin rays. The obtained material was deposited in the Fish Collection of the Federal University of Tocantins.

The abundance of eggs and larvae was calculated per 10 cubic meters of filtered water, according to the method of [9], as modified by [8].

The data were organized according to the dates and locations of sampling, as well as the sampling point (margin/medium; surface), and stored in two .xls files. One of these files contained abiotic data about the environment, and the other file stored biotic data about the eggs and larvae collected.

4 Methodology for Data Mining

The first stage of the work consisted of examining those data mining algorithms that can determine patterns among the properties of the data, thus identifying the unknown relationships between them.

Algorithms that are able to find such relationships are called algorithms of association rules, and extract frequent sets of attributes embedded in a larger set. These algorithms vary greatly in terms of their generation of subsets of the universe, and how the sets of chosen attributes are supported for the generation of association rules.

An association rule has the form $A \rightarrow B$, where the antecedent A and consequent B are sets of items or transactions. The rule can be read as: the attribute A often implies B [10]. To evaluate the generated rules, we used various measures of interest. The most often used [10] are the support and confidence. The authors [11] performed a survey of other metrics, and suggested strategies for selecting appropriate measures for certain areas and requirements. In this paper, we use the following measures:

- Support: P (AB). The support of a rule is defined as the fraction of items I that can be placed in sets A and B based on the given rule. If the support is not large enough, the rule is not worthy of consideration, or is simply deprecated and may be considered later.
- Confidence: P (A / B). This is a measure of the strength of a rule's support, and corresponds to statistical significance. It describes the probability of the rule finding B such that the transaction also contains A.
- Lift: P (B | A) / P (B) or P (AB) / P (A) P (B). Used to find dependencies, the lift indicates how much more common B becomes when A occurs. This value can vary between 0 and ∞.

Finding item sets with frequency greater than or equal to the user-specified minimum support is not trivial, as combinatorial explosion occurs when generating subsets of items. However, because frequent item sets are obtained, it is straightforward to generate association rules with confidence greater than or equal to the user-specified minimum value [12].

In this context, the Apriori algorithm is a seminal method of finding frequent item sets. Introduced in [5], and appointed by the IEEE International Conference on Data Mining [12] as the most promising generation algorithm for association rules, Apriori is one of the most popular approaches in data mining.

Using a depth-first search, the algorithm is able to generate sets of candidate items (recognized as the standard) with k elements from item sets of k-1 elements. The scan ends at the last element of the database, and infrequent patterns are discarded. In this paper, we implement the Apriori algorithm using the Waikato Environment for Knowledge Analysis (Weka) tool [13]. This version of Apriori iteratively reduces the minimum support until it finds the required number of rules with some minimum confidence parameter passed by the user. This ease of parameterization, and the fact

that it includes all evaluation metrics of association rules mentioned above, led to the adoption of Weka instead of a new implementation of the algorithm for rule extraction.

4.1 Preprocessing

The data are only considered to be of sufficient quality if they meet the requirements for their intended use. There are many factors that make up the quality of the data, including accuracy, completeness, consistency, timeliness, credibility, and interpretability [14].

To ensure these measures of quality, the following steps were applied as data preprocessing: (a) data integration, (b) data cleanup, (c) data reduction, (d) data transformation.

Data Integration. Ichthyoplankton samples collected at Lajeado reservoir and its area of influence were recorded in two different databases. To reduce redundancy and inconsistencies in the final dataset, we performed a secure integration that used the sample code as a key link between spreadsheets. The final data set was created in a new CSV (comma separated values) file. Redundant data were eliminated or grouped, depending on the value for the sample, avoiding inconsistencies in the set as a whole.

Data Cleaning. At this stage, various routines were performed to ensure data quality:

- Missing data were replaced by a global constant, indicated by "?" Thus, the algorithm could handle gaps without influencing the results.
- Some nonstandard values (outliers), such as values of "9999" for the water temperature, were removed. These data were considered missing, and were subsequently identified by the symbol "?"
- Inconsistent data, such as input "i" for the cloudiness attribute, which should receive only numeric values by default, were also reported as missing.

Reduction of Data. The final set of attributes from the original set was reduced by performing a dimension reduction in which irrelevant, weakly relevant, and redundant properties were detected and removed.

For this task we employed the CfsSubsetEval algorithm, which assesses the value of a subset of attributes by considering the predictive ability of each individual feature, along with the degree of redundancy between them. Subsets of features that are highly correlated with the class and have low intercorrelation are preferentially selected [15].

For this work, the combination of BestFirst (search method) and CfsSubsetEval (attribute evaluator) is as efficient as the best techniques (genetic algorithm, simulated annealing) for the selection of variables, but has the advantage of being faster than other approaches [16].

To evaluate the attributes, we compared values using the heuristic merit of each relationship, which is formalized as [17]:

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\,\overline{r_{ff}}}} \quad (1)$$

The final formula of merit uses the Pearson correlation between a variable composite (sum or average) and a target variable (the class in question) [18].

The Weka CfsSubsetEval algorithm was executed with the initial set of data as input. From the 33 starting attributes, the base was reduced to 14. After being evaluated by experts in ichthyology, this number was further reduced to a total of 10 attributes considered essential for modeling the problem. The list of attributes and their respective scores of merit is shown in Table 1.

Table 1. Attributes selected by the CfsSubsetEval algorithm from Weka. The stage attribute refers to the larval stage of the fish.

Attribute Rated	Merit	Selected Attributes
Family	0.692	order, specie
Order	0.595	Smf, stage, family
waterConductivity	0.557	local, ph
Specie	0.543	local, waterTemperature, family
Ph	0.463	margin,dissolvedOxigen, waterConductivity, order
dissolvedOxigen	0.445	local, margin, airTemperature, ph
Depth	0.319	local, transparency
transparency	0.314	airTemperatura, depth, dissolvedOxigen, ph
Stage	0.268	order, family
Local	0.116	margin,dissolvedOxigen,ph, waterConductivity,família,specie
waterTemperature	0.096	local, airTemperature
airTemperature	0.094	waterTemperature,dissolvedOxigen, transparency
Margemmargin	0.088	local, dissolvedOxigen, ph, waterConductivity
Smf	0.01	local, margin, ph, waterConductivity, specie

Transformation of Data

- Decimal points input as "," were replaced by "." to ensure they were correctly interpreted by Weka.
- Date formats were standardized to "dd/mm/yyyy".
- Some numerical variables were discretized by mapping value ranges to in labels, as shown in Table 2. This enabled the Apriori algorithm, which requires nominal attributes, to be applied.

After completing this preprocessing, the data were gathered in a single file containing 9 attributes and 4913 instances (from the original 33 attributes and 5333 instances).

5 Results

In the experiments, we used different parameter values for the Apriori algorithm to find the best rules involving the attribute stage (stage of fish larvae) and the biotic and abiotic data.

Table 2. Discretized Attributes

Attribute	Values Gathered in Collection	Equivalent Values	Attribute	Values Gathered in Collection	Equivalent Values
Air Temperature	10 < airTemp <= 15	1	Depth	depth > 3 and <= 4	4
	15 < airTemp <= 20	2		depth > 4 and <= 5	5
	20 < airTemp <= 25	3		depth > 5 and <= 6	6
	25 < airTemp <= 30	4		depth > 6	7
	30 < airTemp <= 35	5	Water Conductivity	cond <= a 20	1
	35 < airTemp <= 40	6		cond > 20 and <= 40	2
Water Transparency	transp > 0 and <= 0.5	1		cond > 40 and <= 60	3
	transp > 0.5 and <= 1	2		cond > 60 and <= 80	4
	transp > 1 and <= 1.5	3		cond >80 and <= 100	5
	transp > 1.5 and <= 2	4		cond>100 and<= 120	6
	transp > 2 and <= 2.5	5		cond > 120	7
	transp > 2.5 and <= 3	6	Dissolved Oxigen	oxig > 0 and <= 1	1
	transp > 3	7		oxig > 1 and <= 2	2
Water Temperature	waTemp>10and<=15	1		oxig > 2 and <= 3	3
	waTemp>15and<=20	2		oxig > 3 and <= 4	4
	waTemp>20and<=25	3		oxig > 4 and <= 5	5
	waTemp>25and<=30	4		oxig > 5 and <= 6	6
	waTemp>30and<=35	5		oxig > 6 and <= 7	7
	waTemp>35and<=40	6		oxig > 7 and <= 8	8
PH	ph <= 4.0	1		oxig > 8 and <= 9	9
	ph > 4.0 and <= 6.0	2		oxig > 9 and <= 10	10
	ph > 6.0 and <= 7.0	3		oxig > 10 and <= 11	11
	ph > 7.0 and <= 9.0	4		oxig > 11 and <= 12	12
	ph > 9.0	5		oxig > 12	13
Depth	depth > 0 and <= 1	1			
	depth > 1 and <= 2	2			
	depth > 2 and <= 3	3			

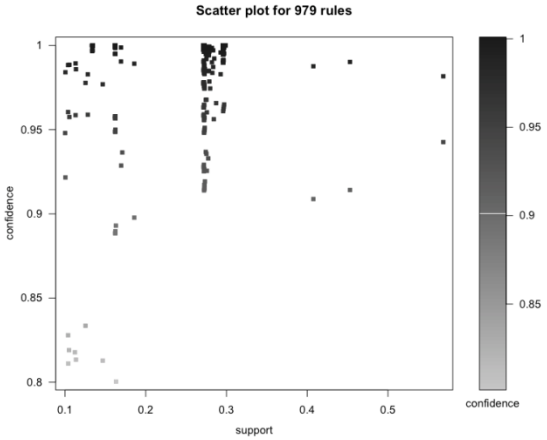
The coverage, or support, of an association rule is taken as the number of instances that are correctly predicted by the rule. Its accuracy, or confidence, is the number of instances that the rule correctly predicts, expressed as a percentage of all instances to which it applies [19].

The minimum and maximum support values were set to 0.1 and 1.0, allowing the rules to be generated freely. Parameter values for the confidence and interest varied

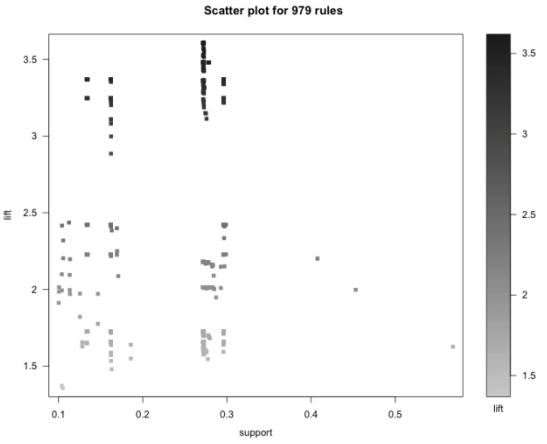
from 0.8 to 1.0 and from 1.5 to 3.5, respectively. In total, 979 association rules were generated, as shown in Graph 1.

A large number of rules were generated with confidence higher than 0.9 and support between 0.2 and 0.3, as shown in Graph 1.

For the metric of interest, the number of rules with lift values between 3 and 3.5 is concentrated between support values of 0.2 and 0.3. Slightly lower lift values, between 1.5 and 2.5, have support in the wider range of 0.1 to 0.3, as can be seen in Graph 2.

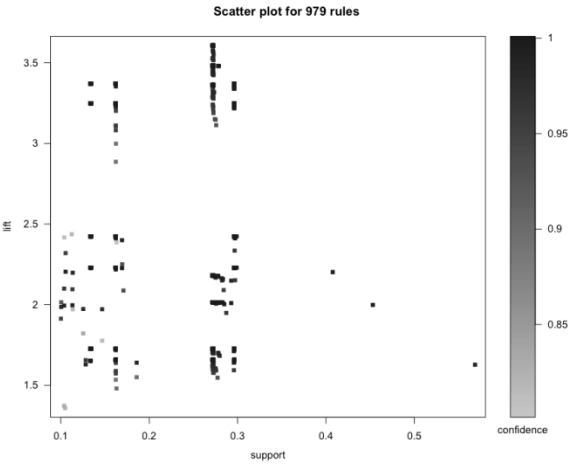


Graph 1. Relation between confidence and support for the 979 generated association rules and their bounds



Graph 2. Number of rules and the relation between lift and support parameters

An analysis of the results showed that rules with a higher confidence value have a support measure of between 0.2 and 0.3, as shown in Graph 3.



Graph 3. Number of rules and the relation between lift, confidence, and support

After analyzing the results, the data mining was refined by considering which of the rules found thus far lack semantic significance, according to experts on fish populations.

The top 10 rules, shown in Chart 1, can thus be interpreted as follows. The value before the symbol "==>" indicates the support of the rule, that is, the number of items covered by a premise(s). The value that appears after the consequent attribute is the number of items for which the consequent of the rule is valid. The confidence value of the rule is given in parentheses. Thus, we can read Rule 1 as follows: "if transparency=6 and pH=5 then depth=7".

1. transparency=6 ph=5 1458 ==> depth=7 1458 conf:(1)
2. dissolvedOxigen=12 transparency=6 1456 ==> depth=7 1456 conf:(1)
3. dissolvedOxigen=12 ph=5 1456 ==> depth=7 1456 conf:(1)
4. dissolvedOxigen=12 ph=5 1456 ==> transparency=6 1456 conf:(1)
5. dissolvedOxigen=12 transparency=6 1456 ==> ph=5 1456 conf:(1)
6. dissolvedOxigen=12 transparency=6 ph=5 1456 ==> depth=7 1456 conf:(1)
7. depth=7 dissolvedOxigen=12 ph=5 1456 ==> transparency=6 1456 conf:(1)
8. depth=7 dissolvedOxigen=12 transparency=6 1456 ==> ph=5 1456 conf:(1)
9. dissolvedOxigen=12 ph=5 1456 ==> depth=7 transparency=6 1456 conf:(1)
10. dissolvedOxigen=12 transparency=6 1456 ==> depth=7 ph=5 1456 conf:(1)

Chart 1. Ten best generated rules.

We can observe that the attribute stage has not been found among the top 10 rules. The attribute stage is included in rules 17 and 22 (Chart 2), which have very high confidence values of 0.99 and 0.96, respectively.

17. transparency=6 stage=pre 524 ==> depth=7 518 conf:(0.99)
22. depth=7 stage=pre 541 ==> transparency=6 518 conf:(0.96)

Chart 1. Association rules with stage attribute

These rules, presented in Chart 2, answer the initial objective of the research, which was to determine the existence of any relationship between abiotic and biotic factors (in this case, the larval stage). The two rules were validated by experts in fish fauna as being important for understanding the process of spawning fish.

6 Final Considerations

The association rules found in this work are consistent with the reality of fish fauna found at the sampling sites, and were semantically validated by ichthyology experts. The data used in this study had previously been analyzed using statistical methods, but no relationships between biotic and abiotic factors were determined. The application of data mining techniques identified new association rules, providing new insights into Amazon fish fauna.

During this research, we found many errors in the input data and/or an inability of existing software to perform the necessary tasks. For the development of future work, and for the application of data mining techniques to other ichthyofauna databases, we are developing specific software to collect data from samples, as well as information visualization routines and decision support systems.

References

1. Hochachka, W.M., Caruana, R., Fink, D., Munson, A., Riedewald, M., Sorokina, D., Kelling, S.: Datamining discovery of pattern and process in ecological systems. Journal of Wildlife Management 71(7), 2427–2437 (2007)
2. Breiman, L.: Bagging predictors. Journal Machine Learning 24, 123–140 (1996)
3. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Journal Machine Learning 36, 105–139 (1999)
4. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York (2009)
5. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994), pp. 487–499 (1994)
6. Agostinho, A.A., Marques, E.E., Agostinho, C.S., Almeida, D.A., Oliveira, R.J., Melo, J.R.B.: Fish ladder of Lajeado Dam: migrations on oneway routes? Neotropical Ichthyology 5(2), 121–130 (2007)
7. Empresa de pesquisa energética – EPE: Plano Decenal de Expansão de Energia 2021. MME/EPE, Brasília (2012)
8. Nakatani, K., Agostinho, A.A., Baumgartner, G., Bialetzki, A., Sanches, P.V., Makrakis, M.C., Pavanelli, C.S.: Ovos e larvas de peixes de água doce: desenvolvimento e manual de identificação. EDUEM. Maringá, 378 p. (2001)
9. Tanaka, S.: Stock assessment by means of ichthyoplankton surveys. FAO Fisheries Technical Paper, vol. 122, pp. 33–51 (1973)
10. Agrawal, R., Imielinski, T., Swami, A.: Database Mining: A Performance Perspective. IEEE Transactions on Knowledge and Data Engineering, Special Issue on Learning and Discovery in Knowledge-Based Databases 5, 914–925 (1993)

11. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys 38(3), Article 9, 9–es (2006)
12. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2007)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1), 10–18 (2009)
14. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco (2011)
15. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Ph.D thesis, Waikato University, Hamilton, NZ (1998)
16. Tetko, I.V., Solov'ev, V.P., Antonov, A.V., Yao, X., Doucet, J.P., Fan, B., Hoonakker, F., Fourches, D., Jost, P., Lachiche, N., Varnek, A.: Benchmarking of Linear and Nonlinear Approaches for Quantitative Structure–Property Relationship Studies of Metal Complexation with Ionophores. Journal of Chemical Information and Modeling 46, 808–819 (2006)
17. Ghiselli, E.E.: Theory of psychological measurement. McGraw-Hill (1964)
18. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: ICML 2000 Proceedings of the Seventeenth International Conference on Machine Learning, pp. 359–366 (2000)
19. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco (2011)

A Pre-initialization Stage of Population-Based Bio-inspired Metaheuristics for Handling Expensive Optimization Problems

Muhammad Marwan Muhammad Fuad

Forskningsparken 3, Institutt for kjemi, NorStruct
The University of Tromsø, The Arctic University of Norway
NO-9037 Tromsø, Norway
mfu008@post.uit.no

Abstract. Metaheuristics are probabilistic optimization algorithms which are applicable to a wide range of optimization problems. Bio-inspired, also called nature-inspired, optimization algorithms are the most widely-known metaheuristics. The general scheme of bio-inspired algorithms consists in an initial stage of randomly generated solutions which evolve through search operations, for several generations, towards an optimal value of the fitness function of the optimization problem at hand. Such a scenario requires repeated evaluation of the fitness function. While in some applications each evaluation will not take more than a fraction of a second, in others, mainly those encountered in data mining, each evaluation may take up several minutes, hours, or even more. This category of optimization problems is called expensive optimization. Such cases require a certain modification of the above scheme. In this paper we present a new method for handling expensive optimization problems. This method can be applied with different population-based bio-inspired optimization algorithms. Although the proposed method is independent of the application to which it is applied, we experiment it on a data mining task.

Keywords: Bio-inspired Optimization, Differential Evolution, Expensive Optimization, Genetic Algorithms, Metaheuristics, Optimization Applications in Data Mining.

1 Introduction

Optimization is an important problem that has many applications. In an optimization problem we try to find a solution that minimizes or maximizes the value of a function that we call the *fitness function* or the *objective function*. Optimization problems can be discrete/ continuous/hybrid, constrained/unconstrained, single objective/ multiobjective, unimodal /multimodal. Optimization algorithms can be classified in several ways one of which is whether they are *single solution –based* algorithms or *population-based* algorithms. The term *metaheuristics* in the optimization literature refers to probabilistic optimization algorithms which are applicable to a large variety of optimization problems. Many of these metaheuristics are inspired by natural

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 396–403, 2013.

processes hence the term *bio-inspired* or *nature-inspired* optimization algorithms. The general scheme used in all these algorithms is the following; an initial stage where a population of feasible solutions is randomly generated. The fitness function of these solutions is evaluated. The solutions with the highest values of the fitness function are favored and are given a higher possibility to survive the optimization process. The algorithm repeats for a certain number of *generations* or *cycles*, or it is terminated by a predefined *stopping criterion*.

As we can see from the above scheme, fitness function evaluation is a central part of bio-inspired optimization. While in some applications each evaluation will not take more than a fraction of a second, in others each evaluation may take up to several minutes, hours, or even more. This category of optimization is called *expensive optimization*. Such cases require a certain modification of the above scheme.

Data mining is a branch of computer science that handles several tasks, most of which demand extensive computing. As with other fields of research, different papers have proposed applying bio-inspired optimization to process data mining tasks [2], [3], [4], [5], [6], [7]. However, most of these applications are expensive optimization problems that require certain considerations.

In this paper we present a new method for handling expensive optimization problems. This method can be applied to different population-based bio-inspired optimization algorithms. Although the proposed method is independent of the application to which it is applied, we test it on a data mining task of setting weights for different segments of time series data according to their information content.

This paper is organized as follows: Section 2 is a background section. In Section 3 we present the new method. The experiments we conducted are reported in Section 4, and we conclude with Section 5.

2 Background

Although bio-inspired algorithms use different search strategies, they all share a common frame that is a based on the following steps:

Initialization: In this step a collection of individuals (called *chromosomes*, *particles*, or *agents*, according to the algorithm) that represent a feasible solution to an optimization problem is generated randomly.

Fitness Function Evaluation: The objective of this step is to rank the different, so-far examined, solutions of the problem, to determine their quality.

Update: The term “update” here does not refer to the narrow meaning of it as it used in *Particle Swarm Optimization* (PSO), but it refers to a meta operation that directs the metaheuristics at iteration $t+1$ towards the region in the search space where a better solution is likely to be found. This update is based on the fitness evaluation at iteration t . This step is the abstract form of the *selection* step used in the *Evolutionary Algorithms* (EA) family.

Mutation: This is a random alteration of a certain percentage of chromosomes. The objective of this operation is to allow the optimization algorithm to explore new regions in the search space.

Iteration: This is not a step by itself, it is the repetition of the last three steps for a predefined number of times (*generations*, *cycles*, *iterations*, depending on the algorithm) which is usually predefined, or until the algorithm is terminated by a stopping criterion.

The performance of bio-inspired optimization highly depends on running the algorithm for a number of iterations sufficient to allow the solutions to evolve, otherwise the algorithm will, at best, only reach a local extreme point. The number of iterations, in turn, is dependent of the computational cost of fitness function evaluation. In general the number of fitness function evaluations can (roughly) be given by:

$$nEval = nItr.sPop \tag{1}$$

Where *nEval* is the number of fitness function evaluations, *nItr* is the number of iterations (generations) and *sPop* is the population size. We say that relation (1) is an approximate one because there are quite a number of variations; for instance, most algorithms will add to that relation another term related to the evaluations resulting from mutation, others will recycle evaluations from previous generations, etc.

One of the trivial techniques to handle expensive optimization problems is simply to reduce the number of generations *nItr*. While this may be acceptable to a certain degree, it could have serious consequences when *nItr* is drastically decreased. Bio-inspired optimization algorithms are supposed to mimic natural phenomena. For instance; EA simulate evolution of species as it happens over thousands of generations. This is the reason why many applications set *nItr* to 1000 or 2000 or even more. But when in some applications of expensive optimization *nItr* is set to 10, for instance, this changes the whole nature of the bio-inspired algorithm. At best, the algorithm in this case can only find a local extreme point, but in other cases the whole optimization process becomes meaningless. Besides, it is important to remember that the random initialization of the population assumes that the algorithm will be run for a certain number of generations enough to “erase” the effect of initialization of the population with specific chromosomes.

3 The Proposed Method

3.1 The Principle

One of the techniques that have already been proposed in bio-inspired optimization to avoid stagnating in a local extreme point is to run the algorithm several times, with different initial populations, and the best result of all these runs is kept. Although this approach is completely inappropriate for expensive optimization problems because it requires too many fitness function evaluations, our method stems from a similar idea;

instead of running the algorithm several times, which is not computationally feasible, and instead of running the algorithm once for a limited number of iterations for expensive optimization problems, as has previously been discussed in Section 2, we propose a new method that runs the algorithm for a limited number of iterations, but using an optimally initialized population.

3.2 Optimization of the Initial Population

As mentioned earlier, our method is based on running an expensive optimization algorithm for a small number of iterations but using an optimally-chosen initial population. However, we should keep in mind that this "optimality" of the initial population should not be determined by any evaluation of the expensive fitness function, otherwise the method would not make sense. The direct result of this requirement is that optimization of the initial population will be problem-independent. To put it simply; we have two separate and independent optimization problems; one is a sub-optimization problem, which is the problem of optimizing the initial population, we call this problem the *secondary optimization problem* and refer to it with (*SecOptim*), and the other is the original optimization problem with the expensive fitness function. We call this problem the *main optimization problem* and we refer to it with (*MainOptim*). *MainOptim* starts the optimization process with an optimal initial population obtained through *SecOptim*.

As a fitness function of *SecOptim* we choose one that gives as much information as possible about the search space of *MainOptim* since this initial population will eventually be used to optimize *MainOptim*. This choice of our fitness function for *SecOptim* originates from one of the rules on which PSO is based, which is the rule of *separation* [8]. According to this rule each particle should avoid getting too close to its neighbors. The intuition behind this rule is that when two particles are close it is very likely that the value of the fitness function for both of them will not be very different. Based on the same intuition, between two different populations we have to choose the one whose chromosomes are as scattered as possible because such a population will give a better representation of the search space. Thus our choice for the fitness function for *SecOptim* will be the one that maximizes the average distance of the chromosomes of the population, i.e.:

$$f_{secOptim} = \frac{2}{secPopSize\,(secPopSize - 1)} \sum_{i=1}^{secPopSize-1} \sum_{j=i+1}^{secPopSize} d(ch_i, ch_j) \qquad (2)$$

where *secPopSize* is the population size of *SecOptim* , *ch* is the chromosome. *d* is a distance, which we choose to be the Euclidean distance. Notice that $d(ch_i, ch_j) = d(ch_j, ch_i)$ so we only need to take half of the summation in (2).

The other component of *SecOptim* is the search space. As indicated earlier, *SecOptim* is a separate optimization problem from *MainOptim* with its own search space. The search space of *SecOptim* is a discrete one whose points are feasible solutions of *MainOptim*. In other words, the search space of *SecOptim* is a *pool* of solutions of *MainOptim* . The cardinality of this pool is denoted by *poolSize*.

Now all the elements of *SecOptim* are defined. *poolSize* is a new element that is particular to our method. In the experimental section we discuss this element further.

3.3 The Algorithm

Briefly, our method as described in Section 3.1 and 3.2, adds to the original optimization problem *MainOptim* another optimization problem *SecOptim* the outcome of which is the initial population of *MainOptim* . The aim of this process is to reduce the number of fitness function evaluations of *MainOptim* by starting the optimization process with an optimal initial population.

4 Application - Experiments

In this section we show how our algorithm is applied through an example of an optimization problem with an expensive fitness function. First we will present the problem and then we will discuss how our method is applied to it, and in the final part of this section we will conduct experiments to test our method.

4.1 The Problem

A *time series* is a collection of observations at intervals of time points. One key to mining time series data is to reduce their dimensionality so that they can be handled efficiently and effectively. Most time series data mining tasks require calculating the similarity between the time series. This similarity is quantified using a similarity measure or a distance metric. In [7] we presented a new distance of time series data, *WPAAD*, which is defined as:

$$WPAAD(S,R) = \sqrt{\frac{n}{N}}\sqrt{\sum_{i=1}^{N} w_i\left(\overline{s_i} - \overline{r_i}\right)^2}; \qquad w_i \in [0,1] \tag{3}$$

Where n is the length of the time series, N is the number of frames in the reduced space, and where the time series are segmented into equal-sized frames, $\overline{s_i}$ ($\overline{r_i}$) is the mean of the data points S (R) that lie within that frame. The weights in (3) are set using the differential evolution (DE) which we present later in this paper. We called the dimensionality reduction technique based on this distance the *Differential Evolutionary Weighted Piecewise Aggregate Approximation* (*DEWPAA*)

4.2 Our Proposed Algorithm

The problem we presented in Section 4.1 is an example of expensive optimization problems, so we will use our algorithm, which we call the *PreInitialAlgo* to show that by starting the optimization process of *DEWPAA* with an optimized population

resulting from our method, we can get the same results of *DEWPAA* but by a much smaller number of generations, thus with much fewer fitness function evaluations. In the language of our *PreInitialAlgo*, the optimization process of *DEWPAA* is *MainOptim*, and *SecOptim* is the optimization process that yields an optimal initial population for *DEWPAA*. Since *MainOptim* and *SecOptim* are independent, we can use two different optimization algorithms if we wish to, so for our experiments we apply the Genetic Algorithms for *SecOptim* and the differential evolution for *MainOptim*. Figure 1 illustrates how *PreInitialAlgo* is applied. But let us first give a brief description of the Genetic Algorithms and the Differential Evolution.

The Genetic Algorithms (GAs): GAs are widely-known bio-inspired optimization algorithms. GA starts by randomly generating a number of chromosomes. This step is called *initialization*. The fitness function of each chromosome is evaluated. The next step is *selection*. The purpose of this procedure is to determine which chromosomes are fit enough to survive. *Crossover* is the next step in which offspring of two parents are produced to enrich the population with fitter chromosomes. The last element is *Mutation* of a certain percentage of chromosomes.

The Differential Evolution (DE): In DE for each individual, which we call the *target vector* $\vec{T}_i$, of the population we randomly choose three mutually distinct individuals; $\vec{V}_{r1}, \vec{V}_{r2}, \vec{V}_{r3}$ which combine to form the *donor vector* $\vec{D} = \vec{V}_{r1} + F\left(\vec{V}_{r2} - \vec{V}_{r3}\right)$. F is called the *mutation factor* . Then a *trial vector* $\vec{R}$ is formed from elements of $\vec{T}_i$ and $\vec{D}$. This includes utilizing another control parameter C_r called the *crossover constant*. In the next step $\vec{R}$ is compared with $\vec{T}_i$ to decide which one of them will survive in the next generation.

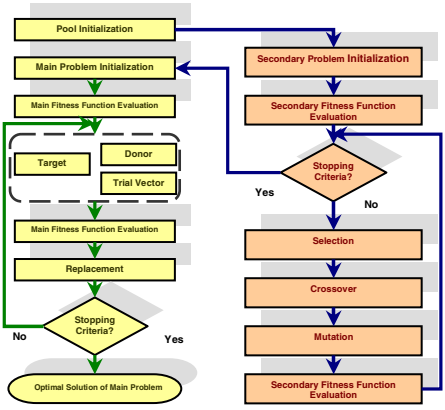


Fig. 1. A scheme of *PreInitialAlgo* using GA and DE

4.3 Experiments

We conducted our experiments on the datasets available at [1] to compare *PreInitialAlgo* with *DEWPAA*. For each tested data set we ran *PreInitialAlgo* for 20 generations to get the weights w_i in relation (3) that minimize the classification error of the training datasets, and then we used these optimal values w_i to classify the corresponding testing datasets to get the classification error. We repeated this process for three compression ratios 1:8, 1:12, and 1:16. We then ran *DEWPAA* for 100 generations to get w_i , also for the same compression ratios. The experiments were conducted on Intel

Core 2 Duo CPU with 3G memory. We present in Table 1 the results of our experiments. As we can see in Table 1 the classification error of *PreInitialAlgo* is equal to, or even better than, that of *DEWPAA* even though the former is run only for 20 generations while the latter is run for 100 generations, which means that *DEWPAA* requires 5 times more fitness function evaluations than *PreInitialAlgo*, yet its performance is the same, or even not as good, as that of *PreInitialAlgo*.

Table 1. Comparison of classification accuracy between *PreInitialAlgo* and *DEWPAA* on different datasets for compression ratios 1:8, 1:12, and 1:16

Dataset	Method	Compression Ratios		
		1:8	1:12	1:16
Lighting7	PreInitialAlgo	0.397	0.260	0.479
	DEWPAA	0.438	0.384	0.479
MedicalImages	PreInitialAlgo	0.378	0.337	0.387
	DEWPAA	0.379	0.353	0.378
Lighting2	PreInitialAlgo	0.213	0.180	0.131
	DEWPAA	0.213	0.197	0.197
MALLAT	PreInitialAlgo	0.095	0.077	0.082
	DEWPAA	0.094	0.094	0.094
FaceUCR	PreInitialAlgo	0.240	0.302	0.364
	DEWPAA	0.238	0.316	0.366
FISH	PreInitialAlgo	0.194	0.246	0.200
	DEWPAA	0.194	0.240	0.229
synthetic_control	PreInitialAlgo	0.063	0.110	0.147
	DEWPAA	0.053	0.113	0.160

The experiments we conducted also included wall clock time comparison. We present in Table 2 the run time of the experiments presented in Table 1. As we can see from the results presented in Table 2, *PreInitialAlgo* is on average 5 times faster than *DEWPAA*, yet the classification errors of both methods are the same in general, which proves the effectiveness of *PreInitialAlgo*.

The results presented in Table 1 and Table 2 were those for *poolSize* =1000. We conducted other experiments for different values of *poolSize* higher than that, and the results were similar.

An interesting thing to mention is that we computed the wall clock time of *SecOptim;* it took only between 7-12 seconds, which is very small compared to the optimization process of *MainOptim*, so this additional secondary optimization

Table 2. Run time comparison between *PreInitialAlgo* and *DEWPAA* for the experiments presented in Table 1

Dataset	Method	Compression Ratios		
		1:8	1:12	1:16
Lighting7	PreInitialAlgo	00h 19m 59s	00h 13m 20s	00h 09m 34s
	DEWPAA	01h 36m 29s	01h 11m 20s	00h 51m 08s
MedicalImages	PreInitialAlgo	03h 09m 23s	02h 03m 36s	01h 39m 58s
	DEWPAA	16h 35m 59s	11h 25m 40s	08h 48m 33s
Lighting2	PreInitialAlgo	00h 32m 12s	00h 21m 56s	00h 17m 00s
	DEWPAA	02h 30m 13s	01h 31m 27s	01h 16m 06s
MALLAT	PreInitialAlgo	00h 43m 19s	00h 29m 30s	00h 18m 42s
	DEWPAA	03h 30m 42s	02h 08m 55s	01h 37m 23s
FaceUCR	PreInitialAlgo	01h 04m 04s	00h 41m 55s	00h 32m 33s
	DEWPAA	05h 31m 19s	03h 27m 50s	02h 49m 21s
FISH	PreInitialAlgo	02h 57m 03s	02h 47m 01s	01h 53m 45s
	DEWPAA	16h 12m 54s	11h 23m 10s	07h 23m 10s
synthetic_control	PreInitialAlgo	01h 04m 47s	00h 58m 27s	00h 30m 11s
	DEWPAA	05h 56m 10s	04h 01m 09s	03h 12m 27s

problem we added did not require but a very small additional computational cost, yet the gain was high. (The wall clock time of *SecOptim* is independent of the dataset, since, as we mentioned earlier, *SecOptim* is independent of *MainOptim*)

5 Conclusion

We presented in this paper a new method, *PreInitialAlgo*, for handling expensive optimization problems such as those encountered in data mining. The new method is applied to population-based bio-inspired algorithms. The basis of our method is to start the optimization process using an optimal initial population. This optimal population is the outcome of another, secondary optimization problem, which is independent of the original problem. We showed experimentally how our new method can substantially improve the performance of the optimization algorithm in terms of speed.

In this paper we used DE and GA as optimizers for the main and secondary optimization problems, respectively. As future work, we would like to test different combinations of bio-inspired algorithms to see which two methods can work best together to yield the best results.

Another direction of future work is to apply the secondary problem using a different fitness function, which could give better results yet.

References

1. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering Homepage (2011), http://www.cs.ucr.edu/~eamonn/time_series_data/
2. Muhammad Fuad, M.M.: ABC-SG: A New Artificial Bee Colony Algorithm-Based Distance of Sequential Data Using Sigma Grams. In: The Tenth Australasian Data Mining Conference - AusDM 2012, Sydney, Australia, December 5-7 (2012)
3. Muhammad Fuad, M.M.: Differential Evolution versus Genetic Algorithms: Towards Symbolic Aggregate Approximation of Non-normalized Time Series. In: Sixteenth International Database Engineering & Applications Symposium – IDEAS 2012, Prague, Czech Republic, August 8-10 (2012)
4. Muhammad Fuad, M.M.: Genetic Algorithms-Based Symbolic Aggregate Approximation. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 105–116. Springer, Heidelberg (2012)
5. Muhammad Fuad, M.M.: Particle Swarm Optimization of Information-Content Weighting of Symbolic Aggregate Approximation. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS (LNAI), vol. 7713, pp. 443–455. Springer, Heidelberg (2012)
6. Muhammad Fuad, M.M.: Towards Normalizing the Edit Distance Using a Genetic Algorithms–Based Scheme. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS (LNAI), vol. 7713, pp. 477–487. Springer, Heidelberg (2012)
7. Muhammad Fuad, M.M.: Using Differential Evolution to Set Weights to Segments with Different Information Content in the Piecewise Aggregate Approximation. In:16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2012. Frontiers of Artificial Intelligence and Applications (FAIA), San Sebastian, Spain, September 10-12. IOS Press (2012)
8. Reynolds, C.W.: Flocks, Herds and Schools: A Distributed Behavioral Model. SIGGRAPH Comput. Graph. 21, 4 (1987)

A Hybrid-Sorting Semantic Matching Method

Kan Li, Wensi Mu, Yong Luan, and Shaohua An

School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
likan@bit.edu.cn

Abstract. Semantic Web Service enables automated service discovery, and even execution. Discovering proper services which match the given request has become an important issue as Internet technology develops and service demands increase. Existing service matching methods in service discovering always cost too much time for obtaining precision and recall ratio. This paper proposes a semantic similarity computing algorithm which considers both information capacity and ontology concept distance. Based on this similarity computing algorithm, we present a hybrid-sorting web service matching method. Compare with other methods like OWLS-MX by experiments, our method reduce average query time and we even improve recall ratio and precision ratio. In addition, this method supports service composition, another important issue in service discovery.

Keywords: Semantic, similarity computing algorithm, service matching method.

1 Introduction

With the developing of Internet, web services become more useful and popular, web services are plentiful and fine-grained, discovering proper services from a wide variety of services is an important issue in web service research.

Several years ago, services were not well understood and calculated by machines. To make the web and service understandable by machines and to increase the level of automatic web service discovery, composition and execution, W3C Semantic Web working group [12] worked for years and proposed Ontology Web Language for Services (OWL-S) [5]. Existing semantic service matching methods are mostly developed based on OWL-S for OWL supports reasoning on concepts and relations [2].

This paper provides a hybrid-sorting matching method that combines logical reasoning and non-logical similarity to match the given request, we consider text Description and QoS. Our experiments show the precision ration and recall ration are better than other methods, and query time improves obviously.

The remainder of this paper is organized as follows. We briefly introduces the developing and related work of semantic service and existing service matching methods in section 2, and then we present our hybrid-sorting web service matching method in detail in Section 3. In Section 4, we give the experiment results of measuring the performance (include query time, precision ratio and recall ration) comparison with other methods, Section 5 is a short discussion of the future work in semantic web service and concludes with a conclusion.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 404–413, 2013.

2 Related Work

Before OWL-S appears, several semantic-enabled specifications for Web service have also been proposed on top of the industrial standard WSDL [9], e.g, WSDL-S [6,7], SAWSDL [11], etc. To adapt web service's development and automation, semantic web service [15] appears. After years, researchers proposed RDF, DAML [16], OIL and corresponding matching methods and systems [17]. Based on DAML+OIL, W3C proposed OWL, which makes a great contribute to semantic web service.

OWL-S's prevalence leads to some web service matching methods and machines such as OWLS-iMatcher [8,10], OWLS-MX [1], etc. OWLS-MX presents a hybrid matching method which combine logical reasoning and IR-based matching criteria to ensure correctly matching those concepts which are semantically synonymous or very closely related. OWLS-MX particularly relies on performance in information retrieval. S Alhazbi, KM Khan, A Erradi considered user preference [4].

In summary, semantic web service matching is still an unsolved problem which attracts many researchers' attention. We also focus on this issue and propose our own solution, a hybrid-sorting service matching approach.

3 Hybrid-Sorting Web Service Matching Method

This service matching method consists of two parts, IO service matching and service sorting. IO matching includes two steps, first is the logical reasoning matching, some advertisements are annotated with its individual degree of matching and then put into the right sets, and some are annotated with "Failed". Then IO similarity matching may find some advertisements' similarity Sim(I, R)> α (α is a given threshold), which means these advertisements match the request in a way. When sorting services in individual sets, we present two ways: text Description semantic similarity sorting and QoS sorting, these two sorting ways present how well this service can satisfy the request in different metric.

3.1 IO Logical Reasoning Service Matching

Existing logical matching methods define services' match degree as five degrees, which is widely recognized by researchers. In this paper, we continue to use this degree classification [1]. In addition, we define another degree, called Simi, to measure IO similarity matching services. Let T be a concept in domain ontology, $LC(T)$is the set of direct child concepts of T), $NLC(T)$ is the set of non-direct child nodes of T, $Ancestor(T)$ is the set of ancestor nodes of T, $LF(T)$ is the set of direct parent concepts of T.

Exact: For $\forall \mathrm{IR}_i \in R, i \in (1,2,\dots,n) \Rightarrow \exists IS_j, j \in (1,2,\dots,m) \cap \mathrm{IR}_i = \mathrm{IS}_j$ and for $\forall \mathrm{OR}_i \in R, i \in (1,2,\dots,n) \Rightarrow \exists OS_j, j \in (1,2,\dots,m) \cap \mathrm{OR}_i = \mathrm{OS}_j$. The service I/O signature perfectly matches with the request with respect to logical equivalence of their formal semantics.

Plug-In: For $\forall \mathrm{IR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists IS_j, j \in (1,2,\ldots,m) \cap \mathrm{IS}_j \in \mathrm{Ancestor}(\mathrm{IR}_i)$, (and $\forall \mathrm{OR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists OS_j, j \in (1,2,\ldots,m) \cap \mathrm{OS}_i = \mathrm{LC}(\mathrm{OR}_j)$.

Subsumes: For $\forall \mathrm{IR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists IS_j, j \in (1,2,\ldots,m) \cap \mathrm{IS}_j \in \mathrm{Ancestor}(\mathrm{IR}_i)$, and for $\forall \mathrm{OR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists OS_j, j \in (1,2,\ldots,m) \cap \mathrm{OR}_j \in \mathrm{Ancestor}(\mathrm{OS}_i)$.

Subsumed-by: For $\forall \mathrm{IR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists IS_j, j \in (1,2,\ldots,m) \cap \mathrm{IS}_j \in \mathrm{Ancestor}(\mathrm{IR}_i)$, and for $\forall \mathrm{OR}_i \in R, i \in (1,2,\ldots,n) \Rightarrow \exists OS_j, j \in (1,2,\ldots,m) \cup \mathrm{OS}_i \in LF(\mathrm{OR}_j) \cap Sim(\mathrm{OS}_i, \mathrm{OR}_j) > \alpha$ (here $LF(\mathrm{OR}_j)$ is the direct parent concepts of $\mathrm{OR_i}$ in domain ontology and α is given threshold value).

Simi: $Sim(\mathrm{OS}, \mathrm{OR}) > \alpha$, α is a given threshold.

Failed: Service S fails to match with request R.

Before we begin the matching, we use OWLS-API [18] to analyze advertisements and read their information (hasInputs, hasOutputs, textDescription and QoS) and store them in a database.

For given request R, we will get vector IR={$\mathrm{IR_1}$, $\mathrm{IR_2}$, …, $\mathrm{IR_n}$} from hasInputs and OR={$\mathrm{OR_1}$, $\mathrm{OR_2}$,…, $\mathrm{OR_m}$} from hasOutputs.

Step 1: Read a piece of data from database, analyze Inputs and Outputs and get IS={$\mathrm{IS_1}$, $\mathrm{IS_2}$,…, $\mathrm{IS_n}$} and OS={$\mathrm{OS_1}$, $\mathrm{OS_2}$,…, $\mathrm{OS_m}$}.

Step 2: For each $\mathrm{IR_i}$ in IR：

(a) If $\mathrm{IR_i}$=Φ, randomly choose $\mathrm{IS_j}$, annotate its match-degrees $\mathrm{md_{Ij}}$ with "Exact" and delete $\mathrm{IS_j}$ in IS; if $\mathrm{OR_i}$≠Φ, turn to (b).

(b) Check if exists $\mathrm{IS_j}$=$\mathrm{IR_i}$ in IS by ontology reasoner like Jena and Pellet, annotate $\mathrm{IS_j}$ match-degree $\mathrm{md_{Ij}}$ with "Exact" and delete $\mathrm{IS_j}$ in IS; if not exist, turn to (c).

(c) Check if exists $\mathrm{IS_j} \in$ Ancestor ($\mathrm{IR_i}$), annotate $\mathrm{IS_j}$ match-degree $\mathrm{md_{Ij}}$ with "Plug-In" and delete $\mathrm{IS_j}$ in IS; if not exist, annotate $\mathrm{IS_j}$ match-degree $\mathrm{md_{Ij}}$ with "Failed" and exit logical reasoning IO matching.

By above steps, we have Inputs matching-degree vector ISM={$\mathrm{md_{I1}}$, $\mathrm{md_{I2}}$, …, $\mathrm{md_{In}}$} ($\mathrm{md_{Ii}}$ is match-degree of each $\mathrm{IS_i}$) for this advertisement, take

$$\mathrm{MD_{IR}} = \min\{\mathrm{md_{Ii}}\} \tag{1}$$

as the Inputs match-degree of this advertisement, $\mathrm{md_{Ii}}$ is match-degree of each $\mathrm{IS_i}$.

The match-degrees' level are defined as: Φ>Exact> Plug-In> Subsumes> Subsumed-by> Fail.

Step 3: For each $\mathrm{OR_i}$ in OR：

(a) If $\mathrm{OR_i}$=Φ, randomly choose $\mathrm{OS_j}$ and annotate its match-degree $\mathrm{md_{Oj}}$ with "Exact", delete $\mathrm{OS_j}$ in OS; if $\mathrm{OR_i}$≠Φ, turn to (b).

(b) Check if exists $\mathrm{OS_j}$=$\mathrm{OR_i}$, annotate $\mathrm{OS_j}$ match-degree $\mathrm{md_{Oj}}$ with "Exact" and delete $\mathrm{OS_j}$ in OS; if not exist, turn to (c).

(c) Check if exists $\mathrm{OS_j} \in$ LC($\mathrm{OR_i}$) annotate $\mathrm{OS_j}$ match-degree $\mathrm{md_{Oj}}$ with "Plug-In" and delete $\mathrm{OS_j}$ in OS; if not exist, turn to (d).

(d) Check if exists $OS_j \in NLC(OR_i)$, annotate OS_j match- degree md_{Oj} with "Subsumes" and delete OS_j in OS; if not exist, turn to (e).

(e) Check if exists $OS_j \in$ Ancestor (OR_i), compute pairwise similarity between Ancestor(OR_i) and OR_i ($Sim(OR_i, OS_j)$), if $Sim(OR_i, OS_j) > \alpha$(α is a given threshold), annotate OS_j match-degrees md_{Oj} with "Subsumed-by" and delete OS_j in OS; if not exist, annotate OS_j match-degree md_{Oj} with "Failed" and exit IO matching.

By above steps, we have Outputs matching-degree vector OSM={md_{O1}, md_{O2}, …, md_{Om}} (md_{Oi} is match-degree of each OS_i) for this service, take

$$MD_{OR} = \min\{md_{Oi}\} \tag{2}$$

as the Outputs match-degree of this advertisement.

Step 4: We describe the degree by two-tuple (MD_{IR}, MD_{OR}). The terminal result of logical reasoning MD_R is the general match-degree, which can be got as follows:

$$MD_R = \begin{cases} \text{Subsumed} - \text{by} & while(MD_{IR}, MD_{OR}) = (Plug - In, Exact) \\ \min(MD_{IR}, MD_{OR}) & others \end{cases} \tag{3}$$

According to MD_R, we have five service sets: Exact, Plug-In, Subsumes, Subsumed-by and Failed (we will use another matching method to deal with Failed set).

Step 5: Repeat Step 1, Step 2, Step 3 and Step 4 until we walk through the database.

After the above processes, all the advertisements in database are annotated with their individual match-degrees.

3.2 IO similarity Service Matching

Similarity Algorithm

Logical reasoning based on subsumption relations of ontology concepts, it's a good way to get concepts relations, however, some other relations can describe similar concepts in ontology, but logical matching method is not able to deal with the situations. We use semantic similarity matching to complement the logical reasoning matching. We compute similarities between advertisements and request R in logical Failed set and put those whose similarity is larger than α (α is a given threshold) into another set called Simi. Advertisements in Simi set also match the request in some extent.

Key to semantic IO similarity service matching is a proper similarity computing algorithm. The mainstream of existing algorithms can be summarized as two basic types: similarity computing based on geometrical distance and similarity computing based on information capacity.

However, both geometrical distance and information capacity based similarity computing methods consider only one aspect of the ontology, this limits the accuracy of similarity. We propose a method that combines geometrical distance and information capacity based similarity computing method, this method considers both distance and information capacity, provide a better measurement of concept similarity in ontology.

Before we present this algorithm, let's see some useful definitions.
Def 1:

$$\mathrm{Freq(n)} = \mathrm{freq(n)} + \sum_{w \in \mathrm{Child(n)}} \mathrm{freq(w)} \tag{4}$$

Freq(n) is the sum of frequency of n and n's children. freq(w) is a concept's frequency of occurrence. Child(n) is the set of all the children of concept n.
Def 2: Edge Energy: Similarity between two concepts in ontology:

$$\mathrm{e}\left(\mathrm{n_c}, \mathrm{n_p}\right) = \frac{\mathrm{Freq(n_c)}}{\mathrm{Freq(n_p)}} \tag{5}$$

$\mathrm{e}\left(\mathrm{n_c}, \mathrm{n_p}\right)$ is the similarity between father concept and child concept in ontology.
Def 3: Total Edge Energy: Sum of the edge energy of the shortest path from node to node:

$$\mathrm{E}(\mathrm{n_1}, \mathrm{n_2}) = \sum_{\mathrm{n_c}, \mathrm{n_p} \in \mathrm{Path}(\mathrm{n_1}, \mathrm{n_2})} \mathrm{e}\left(\mathrm{n_c}, \mathrm{n_p}\right) \tag{6}$$

So the semantic similarity calculating formula:

$$\mathrm{Sim}(\mathrm{n_1}, \mathrm{n_2}) = \frac{2 * \mathrm{depth}(\mathrm{LCA}(\mathrm{n_1}, \mathrm{n_2}))}{\mathrm{depth}(\mathrm{n_1}) + \mathrm{depth}(\mathrm{n_2})} * \frac{\mathrm{E}(\mathrm{n_1}, \mathrm{n_2})}{\mathrm{path}(\mathrm{E}(\mathrm{n_1}, \mathrm{n_2}))} \tag{7}$$

$\mathrm{Sim}(\mathrm{n_1}, \mathrm{n_2})$ is the similarity between concept $\mathrm{n_1}$ and $\mathrm{n_2}$, $\mathrm{depth}(\mathrm{LCA}(n_1, n_2))$ represents the depth of the least common ancestor of $\mathrm{n_1}$ and $\mathrm{n_2}$ $((\mathrm{LCA}(n_1, n_2)), \mathrm{depth}(n_1)$ is the depth of $\mathrm{n_1}$, $\mathrm{depth}(n_2)$ is the depth of $\mathrm{n_2}$ (here the depth of root is 1), $\mathrm{path}(n_1, n_2)$ is the shortest distance between concept $\mathrm{n_1}$ and $\mathrm{n_2}$.

IO Similarity Service Matching

With the proposed similarity computing method, we separately compute Inputs similarity, $\mathrm{Sim}(\mathrm{I_R}, \mathrm{I_S})$ and Outputs similarity $\mathrm{Sim}(\mathrm{O_R}, \mathrm{O_S})$ between request and advertisement, and then give them weights $\mathrm{w_1}$ and $\mathrm{w_2}$ by experience, the general similarity between request and advertisement can be calculated by:

$$\mathrm{Sim}(\mathrm{R}, \mathrm{S}) = \mathrm{w_1}\mathrm{Sim}(\mathrm{I_R}, \mathrm{I_S}) + \mathrm{w_2}\mathrm{Sim}(\mathrm{O_R}, \mathrm{O_S}) \quad (\mathrm{w_1} + \mathrm{w_2} = 1) \tag{8}$$

For request and advertisement's Inputs and Outputs are not always consistent in order, we need to do something when computing similarity, for example:

Request R's Inputs $\mathrm{IR} = \{\mathrm{IR_1}, \mathrm{IR_2}, \ldots, \mathrm{IR}_n\}$, and advertisement S's Inputs $\mathrm{IS} = \{\mathrm{IS_1}, \mathrm{IS}, \ldots, \mathrm{IS}_m\}$, we compute each $\mathrm{IR_i}$ and $\mathrm{IS_j}$'s similarity, get similarity matrix like this:

$$\mathrm{Sim}(\mathrm{IR}, \mathrm{IS}) = \mathrm{IR} \times \mathrm{IS} = \begin{pmatrix} \mathrm{IR_1IS_1} & \cdots & \mathrm{IR_1IS_m} \\ \vdots & \ddots & \vdots \\ \mathrm{IR_nIS_1} & \cdots & \mathrm{IR_nIS_m} \end{pmatrix} \tag{9}$$

$\mathrm{IR_iIS_j}$ is the similarity of $\mathrm{IR_i}$ and $\mathrm{IS_j}$ $\mathrm{Sim}(\mathrm{IR_iIS_j})$.

Go through the similarity matrix row-by-row, take $\max(\mathrm{IR_1IS_a}), a \in (1,2,\ldots,m)$ (the maximum value of the first row) as the first element of Inputs-similarity vector and delete all the elements in column a, then take $\max(\mathrm{IR_2IS_b}), b \in (1,2,\ldots,m)$ as the second element, delete all the elements in column b in turn. Repeat until the matrix is empty. This way, we will get similarity of each element in IR and its counterpart in IS $\{\max(\mathrm{IR_1IS_a}), \max(\mathrm{IR_2IS_b}), \ldots, \max(\mathrm{IR_nIS_k})\}^T$. So the Inputs-similarity of request and matching service can be calculated by:

$$\mathrm{Similarity}(IR, IS) = \min\{\max(\mathrm{IR_1IS_a}), \max(\mathrm{IR_2IS_b}), \ldots, \max(\mathrm{IR_nIS_k})\}^T \quad (10)$$

Outputs similarity Similarity(OR , OS) can be calculated as the Inputs similarity. By using Formula (8), we can get IO similarity Similarity(I, R).

By this similarity matching, we put some services into Simi set.

3.3 Service Sorting

IO matching puts advertisements in dataset into six degree sets, services in five of them match the request in some extent, the other is "Failed". Each of the five matched sets may contain many services, these services' matching degrees are the same in logic, but their similarities with request are not exactly the same, and their qualities of service (QoS) are different, either. When we show all these five service sets, users may have troubles in choosing service. To make it user friendly, we provide two methods for service sorting, include description information sorting and QoS sorting to be chose, thus, users can easily select which service they want by their own individual requirement.

Description Information Similarity Sorting

Description information describes what this service can do in detail. Read service's text Description, then identify parts of speech using Wordnet [13], and extract key words that can express this sentence clearly and concisely. After ignoring adjectives, conjunctions and etc, we get nouns, verbs and adverbs sequence. Calculate pairwise words similarity between request and advertisement by JiangAndConrath (a self-taking similarity computing method in Wordnet), then we can get a similarity matrix.

For example, request's T1={x_1 , x_2 ,…, x_n} and matching service noun sequence T2={ y_1 , y_2 ,…, y_n }:

$$\mathrm{Sim}(\mathrm{T_1}, \mathrm{T_2}) = \mathrm{T_1} \times \mathrm{T_2} = \begin{pmatrix} \mathrm{x_1y_1} & \cdots & \mathrm{x_1y_m} \\ \vdots & \ddots & \vdots \\ \mathrm{x_ny_1} & \cdots & \mathrm{x_ny_m} \end{pmatrix} \quad (11)$$

$\mathrm{x_iy_j}$ is the similarity of x_i and y_j.

By the same "taking max and delete" method mentioned in IO similarity service matching, we can get $\{\max(\mathrm{x_1y_i}), \max(\mathrm{x_2y_j}), \ldots, \max(\mathrm{x_{1n}y_k})\}^{\mathrm{T}} (\mathrm{i} \neq \mathrm{j} \neq \mathrm{k})$, to simplify the computation, we call it $\{\max_1, \max_2, \ldots, \max_n\}^{\mathrm{T}}$, then the following formula can be used to calculate a word sequence's similarity:

$$\mathrm{Sim_{text}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha_i \max_{\mathrm{i}} \quad (12)$$

When considering weight α_i, we use TF-IDF [14] (Term Frequency & Inverse Documentation Frequency) to dynamically compute each α_i and compute the similarity.

$$w_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log\left(N/n_{ij}\right) \tag{13}$$

t_j represents feature (here it's a word), tf_{ij} represents the frequency of occurrence of t_j in document d_i. N is the number of total documents and n_j is the number of documents contain t_j and idf_j means the reciprocal of t_j.

QoS Sorting

QoS is the quality of service, include time, cost, reliability, security, access and so on. Sometimes when matching services, we may want better quality, then QoS sorting can be chose. A service may have many QoS parameters, and they may have different units of measure, this will cause problems when compare values, so we need to normalize QoS.

We take $S_{Candidate} = \{S_1, S_2, \ldots, S_n\}$ as a candidate advertisements set whose QoS parameters are integer, we assume there are five QoS parameters $Q = \{q_1, q_2, \ldots, q_5\}$, accordingly, we have a matrix, row represents QoS, and column represents service:

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{15} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{n5} \end{pmatrix}$$

Using Formula (14) and (15) to normalize Q, Formula (14) for positive indexes and Formula (15) for negative indexes:

$$Q_{ij} = \begin{cases} \frac{q_{ij} - q_j^{min}}{q_j^{max} - q_j^{min}} & q_j^{max} \neq q_j^{min} \\ 1 & q_j^{max} = q_j^{min} \end{cases} \tag{14}$$

$$Q_{ij} = \begin{cases} \frac{q_j^{max} - q_{ij}}{q_j^{max} - q_j^{min}} & q_j^{max} \neq q_j^{min} \\ 1 & q_j^{max} = q_j^{min} \end{cases} \tag{15}$$

Via Formula (14) and (15), we get a new matrix:

$$Q' = \begin{pmatrix} Q_{11} & \cdots & Q_{15} \\ \vdots & \ddots & \vdots \\ Q_{n1} & \cdots & Q_{n5} \end{pmatrix}$$

Then we can get comprehensive QoS value:

$$QoS(S_i) = \sum_{j=1}^{5} \beta_j * Q_{ij} \tag{16}$$

β_j represents weight of Q_{ij}, the values of β can be given by experience or given by users.

So far, we complete service sorting.

4 Experiments and Results

The experiment data set is owls-tc4, for no standard test collection for OWL-S service retrieval exists yet. Owls-tc4, as the largest data set recognized by many researchers in web service, can be a test data set for web service matching based on owls. Owls-tc4 provides 1083 semantic Web services written in OWL-S 1.1 (and for backward compatibility OWLS 1.0) from nine different domains (education, medical care, food, travel, communication, economy, weapons) [3], and the largest domain set include 359 services.

Our method combines logical reasoning matching and similarity matching, to provide more users satisfied services (improve recall ratio). And when using description information and QoS sorting these services in each set, we try to show users the most satisfied services (improve precision ratio). We do experiments with four hybrid methods OWLS-M1, OWLS-M2, OWLS-M3 and OWLS-M4 in OWLS-MX and our hybrid-sorting method on owls-tc4 dataset. OWLS-M1 uses loss of information based similarity measure, OWLS-M2 uses extended Jaccard similarity measure, OWLS-M3 uses cosine similarity measure and OWLS-M4 uses Jemsen-Shanon divergence based similarity. By experiments, we find that our hybrid-sorting method performs better in Recall- Precision and average query time.

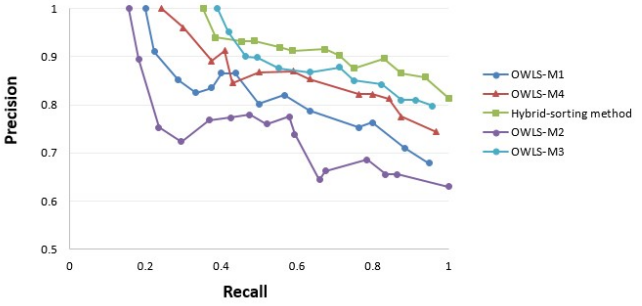


Fig. 1. Retrieval performance

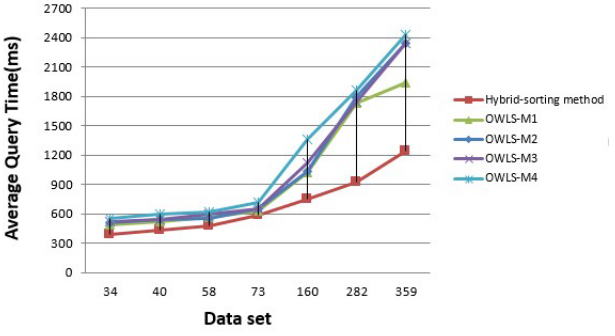


Fig. 2. Average Query Time

Fig. 1 presents the retrieval performance of recall ratio and precise ratio. By this figure, we can see our hybrid-sorting method performs better than other methods in OWLS-MX, even OWLS-M3.

Fig. 2 shows the comparision of average query time between our method and four methods in OWLS-MX. The x axis represents the size of datasets and y axis represents query time (ms). This broken line represents as the size of datasets increase, how query time increases accordingly. Obviously, our hybrid-sorting matching method is much better in query time than OWLS-MX when the size of dataset increases. For the reasons, OWLS-MX use IR when technology which using all the information OWL-S provided, some of them are of little effect, our method abandon those information which effect results little. The most important is, in the previous implementation of SM algorithm, searching is directly operated on disk, which leads highly intensive file access threatening system performance. As an optimized alternative, our solution attempts to hold majority of data access operated on main memory. The SPARQL based searching engine and the well organized data index improve the system performance. Additionally, searching service can be extended for more complex condition, with insignificant overhead.

5 Conclusion

Our hybrid-sorting matching method uses IO functional information to put services into six service sets (by utilizing both logical reasoning and semantic similarity), and we also use non-functional description information and QoS to sort services in different sets. In this way, services are classified and put into six sets, services in five of them match the request, and in each set, services are sorted by users' demands. In this matching method, IO service matching ensures finding all the matching advertisements as far as possible, and by using similarity service sorting, we can give users the most satisfied advertisements. This way, we improve recall ratio, precision ratio and average query time at the same time.

Even more important, this method supports service composition-another issue in service discovery. Based on this matching method, our future work will focus on service composition.

References

1. Klusch, M., Fries, B., Sycara, K.P.: Owls-mx: A hybrid semantic web service matchmaker for owl-s services. Web Semantics 7(2), 121–133 (2009)
2. Masuch, N., Hirsch, B., Burkhardt, M.: Semantic web service, pp. 35–47 (2012)
3. Klusch, M., Khalid, M.A., Kapahnke, P., Fries, B., Vasileski, M.: OWLS-TC, OWL-S Service Retrieval Test Collection Version 4.0-User Manual (2010)
4. Alhazbi, S., Khan, K.M., Erradi, A.: Preference-based semantic matching of web service security policies. In: Computer and Information Technology, WCCIT, pp. 1–5 (2013)
5. Farrell, J., Lausen, H.: Semantic annotations for wsdl and xml schema (2007)

6. Schuldt, H. (ed.): CASCOM - Intelligent Service Coordination in the Semantic Web. Springer, Heidelberg (2008)
7. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.-T., Sheth, A., Verma, K.: Web service semantics - wsdl-s (2005), http://www.w3.org/Submission/WSDL-S/
8. Kiefer, C., Bernstein, A.: The creation and evaluation of iSPARQL strategies for matchmaking. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 463–477. Springer, Heidelberg (2008)
9. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1. W3C Note 15 (2001)
10. Koivunen, M.-R., Miller, E.: W3c semantic web activity. In: Semantic Web Kick-Off in Finland, pp. 27–44 (2001)
11. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., Mcllraith, S., Narayanan, S., Paolucci, M., Parsia, B., Terry, P., Sirin, E., Srinivasan, N., Sycara, K.: Owl-s: Semantic annotated up for web services (2004), http://www.w3.org/Submission/OWL-S/
12. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C recommendation (2004)
13. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38 (1995)
14. Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization (1996)
15. McIlraith, S.A., Martin, D.L.: Bringing semantics to web services. In: IEEE Intelligent Systems, pp. 90–93 (2003)
16. Ankolekar, A., et al.: DAML-S: Web Service Description for the Semantic Web. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 348–363. Springer, Heidelberg (2002)
17. Bansal, S., Vidal, J.M.: Matchmaking of web services based on the DAML-S service model. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (2003)
18. http://www.mindswap.org/2004/owl-s/api/

Improving Few Occurrence Feature Performance in Distant Supervision for Relation Extraction

Hui Zhang and Yuanhao Zhao

School of Computer Science, Beihang University, Beijing 100191, China
{hzhang,zyhzdp}@nlsde.buaa.edu.cn

Abstract. Distant supervision is a hotspot in relation extraction research. Instead of relying on annotated text, distant supervision hires a knowledge base as supervision. For each pair of entities that appears in some knowledge base's relation, this approach find all sentences containing those entities in a large unlabeled corpus and extract textual features to train a relation classifier. The automatic labeling provides a large amount of data, but the data have serious problem. Most features appear only few times in training data, and such insufficient data make these features very susceptible to noise, which will lead to a flawed classifier. In this paper, we propose a method to improve few occurrence features' performance in distant supervision relation extraction. We present a novel model to calculating the similarity between a feature and an entity pair, and then adjust the entity pair' features by their similarity. The experiment shows our method boosted the performance of relation extraction.

Keywords: relation extraction, distant supervision, similarity.

1 Introduction

Relation extraction aims to extract semantic relationship between two or more entities from the text. The traditional supervised approaches are limited in scalability because labeled data is expensive to produce. A particularly attractive approach [1], called distant supervision (DS), creates labeled data by heuristically aligning entities in text with those in a knowledge base, such as Freebase. DS approach does not need any manually labeled data; it uses a knowledge base to create labeled data for relation extraction by heuristically matching entity pairs. Then this approach extracts features from labeled data and combines features of same relation instance and a feature is a sequence of lexical and syntactic information. DS thinks of a feature as a whole and uses features as dimensions to describe data, which leads to high-precise but low-recall.

However, compare to human-labeled data, DS's heuristically labeled data is from large corpus, in which any individual sentence may give an incorrect cue. In training data, more than 90% features' occurrence times are less than 5, if any sentence that contain these features make a mistake, the corresponding feature and the classifier will be affected by noise. For the few occurrence features, more appearing number means more training data, which usually result in more reliable classifier.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 414–422, 2013.

In the testing step of classification problem, more data usually lead to more reliable result, while DS is the same. That's why DS model combines each entity pair's entire sentence to build a richer feature vector. But in fact there are a lot of possible that this gathering is in vain. More than 77% entity pair occurs in only one sentence, which could be extracted only one feature. These once occurrence entity pairs' classification is all dependent on the features contained, so these entity pairs' classification accuracy reflex these features' performance. We find that with the increase in the feature's number of occurrences, the classification accuracy rate is also rising, which means features' more occurrence times lead to more accurate classifier (see section 4).

Previous works focus on DS assumption: If two entities participate in a relation, any sentences that contain those two entities might express that relation, try to remove noisy data introduced by assumption failure.

In this paper, we propose a method to make use of DS features' internal information, while previous works of DS just avoid features' internal information. And we find out that DS's heuristically labeled data could not offer sufficient training data, to solve this problem, we introduce a novel model to calculate the similarity between a feature and an entity pair by features' internal information, and then adjust testing entity pair by its similar features. So a testing entity pair's relation is not only decided by its own features, but also be affected by its similar features. In this way, we reduce the noise introduced by the few occurrence features.

This paper is organized as follow. Section 2 describes the related work. In section 3, we present our model in detail. Section 4 shows experiment result. At last, we conclude experience and outlook in Section 5.

2 Related Work

Relation extraction can be classified according to the degree of human involvement as supervised, semi-supervised, distant supervised and unsupervised methods. Supervised relation extraction methods require large corpus of labeled data, which is often difficult to obtain. [2, 3] utilize a knowledge base to heuristically label a corpus, called distant supervision, or weak supervision. [1] uses Freebase as a knowledge base by making the DS assumption and trained relation extractors on Wikipedia. [4] has pointed out that the DS assumption generates noisy labeled data, but does not directly address the problem. [6] applies a rule-based method to the problem by using popular entity types and keywords for each relation. [5][7] use multi-instance learning, which deals with uncertainty of labels, to relax the DS assumption. [8] models the probabilities of a pattern showing relations, estimated from the heuristically labeled dataset.

3 Few Occurrence Features Improve Model

Our aim is to improve the few occurrence features' robustness to noise, which cause poor precision. Indeed, in our Wikipedia corpus, more than 90% features' occurrence times are less than 5, which means any noise will strongly affect the performance,

which will lead to a significant problem for training. These few occurrence features' training data is so insufficient to build a reliable classifier, and only rely on these features own is not reliable when meet them in testing phrase.

In our Few Occurrence Features Improve Model, we improve the few occurrence features' robustness to noise as follow:

1. Calculate similarity between every feature extracted from training data and every feature from the testing entity pair.
2. Figure out the similarity between every feature extracted from training data and the testing entity pair.
3. Add top n similar features of the testing entity pair and build a richer testing feature vector for the classifier.

3.1 Calculate Similarity between Two Features

We now describe our features' similarity calculation model. A feature in DS consists of the conjunction of several attributes of a sentence, plus the named entity tags. For example, when the named entities 'Edwin Hubble' and 'Marshfield' from sentence 'Astronomer Edwin Hubble was born in Marshfield, Missouri' are recognized, the feature 'Edwin Hubble was born in Marshfield' can be extracted. There are some syntactic and lexical attributes in the feature, the lexical attributes include named entity type 'PERSON' and 'LOCATION' as well as words with POS tags such as 'was/VERB' and 'born/VERB', and the syntactic attributes are shown in Fig. 1. DS uses features as dimensions to describe the data, which ignore features' internal information.

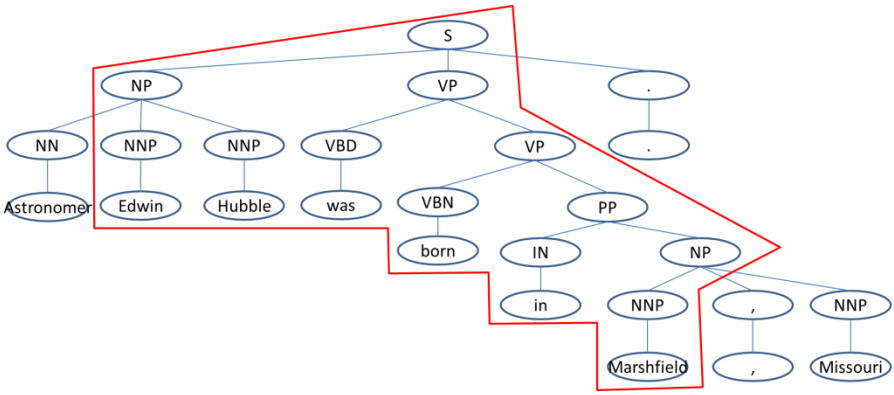


Fig. 1. Syntactic dependency tree

Feature's internal information includes syntactic and lexical attributes, so we can calculate syntactic and lexical similarity between two features. In our model, the similarity between features f1 and f2 is calculated as Eq.1

$$\mathrm{sim}(\mathrm{f1},\mathrm{f2}) = \alpha sim_{syn}(\mathrm{f1},\mathrm{f2}) + (1-\alpha) sim_{lex}(\mathrm{f1},\mathrm{f2}) \quad (1)$$

Where $sim_{syn}(\mathrm{f1},\mathrm{f2})$ is the syntactic similarity between f1 and f2, $sim_{lex}(\mathrm{f1},\mathrm{f2})$ is the lexical similarity between f1 and f2, $\alpha$is the parameter that control syntactic and lexical similarities' ratio.

In order to calculate syntactic similarity, the syntactic attribute of features should be obtained. So we parse each sentence with Stanford Parser to extract a dependency path of the corresponding feature. Then we use convolution tree kernel [9] to calculate similarity between two dependency trees, which is syntactic similarity between two features.

We take a feature as a string, each attribute in the feature as a word. In this way we calculate lexical similarity between two features just like calculate two strings' similarity. Therefore, $sim_{lex}(\mathrm{f1},\mathrm{f2})$ is modeled as

$$sim_{lex}(\mathrm{f1},\mathrm{f2}) = \frac{\mathrm{lcsq}(\mathrm{f1},\mathrm{f2}) * \mathrm{lcss}(\mathrm{f1},\mathrm{f2})}{\mathrm{length}(\mathrm{f2})} \tag{2}$$

Where $\mathrm{lcsq}(\mathrm{f1},\mathrm{f2})$ is the length of longest common subsequence between f1 and f2; $\mathrm{lcss}(\mathrm{f1},\mathrm{f2})$ is the length of longest common substring between f1 and f2; $\mathrm{length}(\mathrm{f2})$ is the length of f2. Note that we take every word in f1 and f2 as a whole, which means there is no half a word in $\mathrm{lcsq}(\mathrm{f1},\mathrm{f2})$ or $\mathrm{lcss}(\mathrm{f1},\mathrm{f2})$.

3.2 Calculate Similarity between a Feature and a Testing Entity Pair

As we could calculate similarity between every two feature, we model the similarity between a feature f and a testing entity pair as follows process:

1. For each feature f' in the testing entity pair e, we can calculate $\mathrm{sim}(\mathrm{f},\mathrm{f}')$ by Eq.1;

2. Then we adjust $\mathrm{sim}(\mathrm{f},\mathrm{f}')$ by

$$\mathrm{sim}'(\mathrm{f},\mathrm{f}') = \mathrm{sim}(\mathrm{f},\mathrm{f}') * \frac{\log(\mathrm{times}(\mathrm{f}) + 2)}{\log(\mathrm{times}(\mathrm{f}') + 2)} \tag{3}$$

where $\mathrm{times}(\mathrm{f})$ is the occurrence times of f in training data. $\mathrm{sim}'(\mathrm{f},\mathrm{f}')$ should be bigger when f is more reliable, but we could not measure how much f is reliable, so we use the occurrence times of f in training data instead of reliable, and f' is the same.

3. Calculate the similarity between f and e by adding up all weights

$$\mathrm{sim}(\mathrm{f},\mathrm{e}) = \sum_{f' \in \mathrm{e}} \mathrm{sim}'(\mathrm{f},\mathrm{f}') \tag{4}$$

Now we have the similarity between every feature and the testing entity pair, we just pick the top m similar features and add them to the testing entity pair's feature vector.

4 Experiments

We performed two sets of experiments.

Experiment 1 aimed to evaluate the performance of our model itself, which reduce noise effect from the DS heuristically labeled data and improve few occurrence features' performance.

Experiment 2 aimed to evaluate how much our few occurrence features improve model in Section 3 improved the performance of relation extraction.

4.1 Dataset

Following Mintz et al. (2009), we carried out our experiments using Wikipedia as the target corpus and Freebase (April, 2013, (Google, 2009)) as the knowledge base. We used more than 4.2 million Wikipedia articles in the dump data (April, 2013). And follow Mintz et al. (2009), we used the exactly same 23 largest relations in Freebase.

In order to find entity mentions in text we extracted entity instances from Freebase. Using these entity instances, POS tag and NER tag, we can find entities in every single sentence from corpus.

Next, for each pair of entities participating in a relation of our training KB, we traverse the text corpus and find sentences in which the two entities co-occur. Each pair of entity mentions is considered to be a relation mention candidate. For each candidate we extract a set of features. The types of features are essentially corresponding to the ones used by DS approach: we used lexical, Part-Of-Speech (POS), named entity and syntactic features (i.e. features obtained from the dependency parsing tree of a sentence). We applied the Stanford POS tagger for POS tags; Stanford named entity recognizer for named entity tags and Stanford Parser for dependency parsing.

The properties of our data are shown in Table 1.

Table 1. Properties of dataset

Property	Value
documents	4,200,000
relation instance in Freebase	5,884,334
matched relation instance in Wikipedia	183,097
features	69,467
relations	24

4.2 Configuration of Classifiers

Following Mintz et al. (2009), we used a multiclass logistic classifier optimized using L-BFGS with Gaussian regularization to classify entity pairs to the predefined 24 relations and NONE. In order to train the NONE class, we randomly picked 100,000 examples that did not match to Freebase as pairs.

4.3 Experiment 1: Few Occurrence Features' Improvement

We compare our model with baseline method in terms of few occurrence features' classification accuracy. As we could not measure every single feature's classification accuracy, we use these one-feature entity pairs' accuracy instead.

Relation instances from Freebase were divided into two parts, half for training and another half for testing. This means that 2.9 million Freebase relation instances are used in training, and 2.9 million are used in testing. The experiment used 2.8 million Wikipedia articles in the training phase and 1.4 million different articles in the testing phase. Note that all entity pairs in training are used to build the classifier no matter how many features are contained, but in testing phase, the entity pairs that only contain one feature were extracted. In this way we can evaluate few occurrence features' classification performance by these entity pairs' classification accuracy.

Result and Discussion

The results of Experiment 1 are shown in Table 2. Our model achieved the best precision in the few (<5) occurrence features' classification. When features' occurrence times in training data is zero, which means these features make no contribution to the classification, in such condition the classification is all decided by the distribution of relations in training data, ending up terrible precision, while our method did a good job. However, with the growth of features' occurrence times, our method's boost to precision gradually decreases.

Table 2. Precision of one feature entity pairs

features' occurrence times in training data	DS	Our model
0	21.34%	54.21%
1	50.67%	65.17%
2	51.82%	57.16%
3	56.24%	57.21%
4	63.79%	66.32%
5	67.40%	69.62%

4.4 Experiment 2: Relation Extraction

Following Mintz et al. (2009), we evaluate labels in two ways: automatically, by holding out part of the Freebase relation data during training, and comparing newly discovered relation instances against this held-out data, and manually, having humans who look at each positively labeled entity pair and mark whether the relation indeed holds between the participants. Both evaluations allow us to calculate the precision of the system for the best N instances.

Held-Out Evaluation

The dataset used in held-out evaluation is exactly the same as the dataset used in experiment 1. Note that in held-out evaluation all entity pairs in testing are extracted no matter how many features are contained.

We calculate the precision of each method for the best n relation instances. The precisions are underestimated because this evaluation suffers from false negatives due to the incompleteness of Freebase. We changed the recall from 0.05 to 1 and measured precision. Precision-recall curves for the held-out data are shown in Fig. 2.

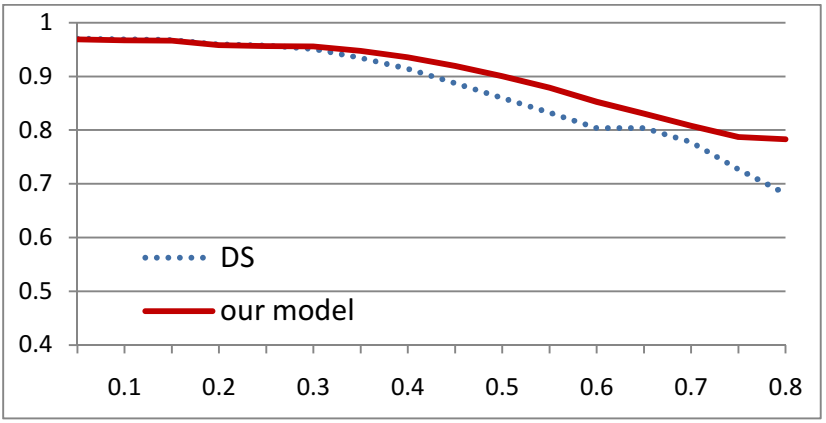


Fig. 2. Precision-recall curves

Our model achieved comparable or higher precision at most recall levels compared with DS approach. Its performance at recall = 0.75 is much higher than that of DS approach.

Manual Evaluation

For manual evaluation all Freebase relation instances are used in training. As candidate relation instances we choose those entity pairs which appear together in Wikipedia test set, but not in any Freebase relation. This means that there is no overlap between the held-out and manual candidates. Then we apply our models to this test set, and manually evaluate the top 50 relation instances for the most frequent 10 relations. The manually evaluation result are shown in Table 3.

Table 3. Estimated precision on human-evaluation experiments of the highest-ranked 50 results per relation

relation name	DS	Our model
/location/location/contains	0.82	0.86
/people/person/place_of_birth	0.56	0.6
/people/person/nationality	0.8	0.8
/book/author/works_written	0.88	0.84
/film/director/film	0.52	0.58
/film/film/genre	0.74	0.74
/people/deceased_person/place_of_death	0.66	0.64
/people/person/profession	0.58	0.66
/film/writer/film	0.58	0.6
/film/film/country	0.16	0.26
average	0.63	0.658

Our model achieved the best average precision, which means our model actually improve DS relation extraction performance.

5 Conclusions and Future Work

This paper shows that few occurrence features of DS model are serious affected by noise because DS heuristically labeled data is insufficient. Then this paper presents a novel approach to improve the robustness of DS to noise by improving few occurrence features' classification performance. In Few Occurrence Features Improve Model, entity pair's relation is determined by its own features as well as its similar features. The experimental results show that this method successfully improves few occurrence features' classification performance and boosted the performance of relation extraction.

Our model makes use of features' internal information and improves DS's performance, which is not used in previous works of DS. If the previous works of DS make full use of features' internal information in their models, their model may achieve better results.

References

1. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL 2009), pp. 1003–1011. Association for Computational Linguistics (2009)
2. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: Proceedings of the 16th ACM International Conference on Information and Knowledge Management (CIKM 2007), pp. 41–50. ACM Press, New York (2007)
3. Bellare, K., Mccallum, A.: Learning extractors from unlabeled text using relevant databases. In: Proceedings of the Sixth International Workshop on Information Integration on the Web (IIWeb 2007), in Conjunction with AAAI 2007, pp. 10–16. AAAI Press, Vancouver (2007)
4. Hoffmann, R., Zhang, C., Weld, D.S.: Learning 5000 relational extractors. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 286–295. Association for Computational Linguistics, Stroudsburg (2010)
5. Riedel, S., Yao, L., McCallum, A.: Modeling Relations and Their Mentions without Labeled Text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010)
6. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining (2010)
7. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 541–550. Association for Computational Linguistics (2011)

8. Takamatsu, S., Sato, I., Nakagawa, H.: Reducing wrong labels in distant supervision for relation extraction. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 721–729. Association for Computational Linguistics (2012)
9. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (2004)
10. Freebase data dumps, http://download.freebase.com/datadumps/

Cluster Labeling Extraction and Ranking Feature Selection for High Quality XML Pseudo Relevance Feedback Fragments Set

Minjuan Zhong*, Changxuan Wan, Dexi Liu, Shumei Liao, and Siwen Luo

School of Information Technology, Jiangxi Key Laboratory of Data and Knowledge Engineering, Jiangxi University of Finance and Economics, Nanchang
lucyzmj@sina.com, wanchangxuan@263.net, Dexiliu@gmail.com, lsmzl@hotmail.com, luosiwen@126.com

Abstract. In traditional pseudo feedback, the main reason of the *topic drift* is the low quality of the feedback source. Clustering search results is an effective way to improve the quality of feedback set. For XML data, how to effectively perform clustering algorithm and then identify good xml fragments from the clustering results is a intricate problem. This paper mainly focus on the latter problem. Based on k-mediod clustering results, This work firstly proposes an cluster label extraction method to select candidate relevant clusters. Secondly, multiple ranking features are introduced to assist the related xml fragments identification from the candidate clusters. Top *N* fragments compose the high quality pseudo feedback set finally. Experimental results on standard INEX test data show that in one hand, the proposed cluster label extraction method could obtain proper cluster key terms and lead to appropriate candidate cluster selection. On the other hand, the presented ranking features are beneficial to the relevant xml fragments identification. The quality of feedback set is ensured.

Keywords: xml search results clustering, cluster label, ranking features, Pseudo Relevance Feedback.

1 Introduction

The Extensible Mark-up Language (XML) has rapidly evolved to an emerging standard for large-scale data exchange and integration over the Internet. The growing number of XML documents lead to the need for retrieval. However, the common problem is that the performance of XML information retrieval is often not satisfactory. Researchers realize that queries, especially short queries, do not provide a complete specification about users' needs. Pseudo relevant feedback(PRF) is an important way to enhance retrieval quality by integrating relevance information provided by the original top N retrieved documents. However, due to system bias, top-ranked documents may contain noise [1], which can ultimately result in the query representation drifting "away" from the original query.

Targeting this limitation and for improving the retrievability of individual documents, in this paper, we propose an method for high quality xml pseudo-relevance

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 423–432, 2013.

feedback fragments selection. The main idea is to use document clustering and ranking scheme to find dominant fragments for PRF. This paper focus on addressing the latter problem. Based on k-mediod xml fragments clustering results, we firstly propose an cluster label extraction method and obtain the cluster key terms to select the candidate cluster. Secondly, multiple ranking features are introduced to assist the related xml fragments identification from the candidate clusters. Top N fragments compose the high quality pseudo feedback set finally.

In the past few years, there were many researches on selecting relevant feedback documents in PRF[2,3]. However, the existing work has been mainly focused on traditional PRF and the treatment object is based on general documents. As far as we are aware, there is little work done XML pseudo relevance feedback and it is almost no research achievements.

Compared with traditional documents, one of the most outstanding features of XML retrieval is that the document components- the so-called XML elements – instead of complete documents, could be return in response to a user query. Therefore, In this paper, our focus is to identifying good XML feedback fragment and forming high quality pseudo feedback set so as to improve the performance of the XML retrieval ultimate. This research makes the following main contributions:

(1)The cluster labeling extraction based on equalization weight is presented. This method could take full consideration in XML dual characteristics(content and structure)and select those terms with high weight value as key terms to reflecting the main idea of one cluster.

(2) We introduce multiple ranking features to assist the related xml fragments selection. The adoption of these features make low quality fragments filter out.

(3)A comprehensive evaluation has been conducted to evaluate the proposed approach. Experiment results show that it is feasible and outperforms strong original query results.

In the remainder of this paper, We first give a brief related works in PRF document selection, then we describe how to find the candidate relevant clusters, in which cluster labeling extraction method is proposed and the cluster key terms are obtained based on k-medoid clustering results, by using them, the candidate clusters is then determined; Section 4 provides multiple ranking features to help relevant xml fragments selection, and in section 5, we report and discuss experimental results on IEEE CS test collections; finally, conclusion are given in the last section.

2 Related Work

As we mentioned in the previous section, current studies are mainly focused on tr aditional PRF and the data object are all general documents. Recently, Ben et al.[2] propose to detect good feedback documents by classifying all feedback docu ments using a variety of features such as the distribution of query terms in th e feedback document, the similarity between a single feedback document and a ll top-ranked documents, or the proximity between the expansion terms and the original query terms in the feedback document. Karthik Raman[3] introduce the

notion of pseudo-irrelevant documents, i.e. high-scoring documents outside of t op n that are highly unlikely to be relevant. Query expansion could be perfor med in the relevant document set through removing the irrelevant documents.

Search results clustering is an another way to identify relevant documents. Sampling and resampling techniques for the initial retrieval results are proposed. A selective sampling method by Sakai et al [4] skips some top-retrieved documents based on a clustering criterion. Kyung et al.[1] describes a resampling method using clusters to select better documents for pseudo-relevance feedback. The main limitation of the approach is that, it is not useful for some application domains. Therefore, [5] proposed an improved approach for clustering, where it checks intra-cluster similarity of clusters on the basis of local frequent terms and rank clusters, and top-ranked documents in high rank clusters for relevance feedback are selected. A resampling method suggested by Collins-Tompson and Callan[6] uses bootstrap sampling on the top-retrieved documents for the query and variants of the query obtained by leaving a single term out.

3 Candidate Relevant Clusters Identification

3.1 Cluster Labeling Extraction Based on Equalization Weight

Cluster labeling could reflect the central idea of one cluster to a certain extent. Hence, we extract the most important terms from the cluster as cluster labeling. Generally, weight may express the importance of the tem in the document. The more important of the term, the larger of the corresponding the weight value is. So, in this paper, the candidate core terms of one document are firstly selected through weight computation formula in [7].

After that, the cluster key terms are chosen based on the following features:

(1) DF(document frequent): refers to the number of document, in which the term is regarded as the core terms. The underlying assumption is that if one candidate term is selected as core terms in many documents of one cluster, then the candidate term is representativeness widely and hence as the label of the cluster.

(2) the average ranking of the candidate key terms . Each candidate key terms have their ranking in each document. One term may rank in the top position, while in the back of the other document. The larger value of the average ranking, the more front of the key term is ranked and the greater contribution to the document is. The specific computation is adopted as following formula:

$$Aver\_RankValue(term_i) = \frac{\sum_{j=1}^{count} Rank\_Value(term_i, p_j)}{count} \quad (1)$$

Where count is the number of the fragments in one cluster, $Rank\_Value(term_i, p_j)$ denotes the ranking of the $term_i$ in fragment p_j which is defined as following equation.

$$Rank\_Value(term, p_i) = (Top\_N - position_i - 1) \Big/ Top\_N \quad (2)$$

The definition of the equalization weight is the product of the DF and the average ranking of the candidate key terms.

$$balance\_weight(term_i) = DF(term_i) * Aver\_RankValue(term_i) \quad (3)$$

The detail algorithm of cluster labeling extraction is described as follows:

Input: XML search results clustering, namely fragment_set

Output: cluster labeling, namely keytermset

1.*for each fragment in fragment_set*

1.1 *fragment=filter_stopword(fragment)// filter the fragment, and remove the stopword;*

1.2 *set={Top_N terms with high weight value}*

1.3 *end for*

2. *unique_set={remove the repetition terms of set}*

2.1 *for each term in unique_set*

2.2 *initial_frequent(unique_set) // initial each different term weight as zero* ;

2.3 *count_df(term, set) // count DF value of term in set* ;

2.4 *computer_averagerank(term) //compute the average ranking of term;*

2.5 *endfor*

3. *Rank(unique_set) // rank term according to the equalization weight and the top_N terms are selected, which save in the keytermset.*

Therefore, the weight of key term ti in the jth cluster is defined as follows:

$$AW(t_i, c_j) = balance\_weight(t_i) * \ln(\frac{N+1}{n_{t_i}}) \quad (4)$$

where N refers to the total number of clusters and n_{ti} is the number of clusters in which t_k is contained as the cluster key terms.

3.2 Candidate Relevant Clusters Ranking Model Based on Cluster Labeling(CRCRM)

The candidate relevant clusters should show much more relevant to the user's query. Therefore, identification of candidate clusters should solve one key problems, that is how to measure the similarity between the cluster and the query.

Through analyzing, it is shown that if one cluster is considered as a good candidate, the relevance degree between the documents in it and user's query intention should be high. Therefore, a naturally idea is to investigate relevance degree between each document of each cluster and the query. However, low efficiency makes it

unrealistic, especially to the large number of clustering documents. So, the similarity computation is firstly performed between each cluster center and user's query, and then rank them in descent according to the similarity values; finally, the top N clusters are regarded as the relevant candidate clusters.

As mentioned in the previous section, cluster labeling as key terms could reflect the main idea of one cluster. Therefore, the cluster center is replaced by the key terms and the similarity is computed between the key term set and queries terms. The formula are shown as follows:

$$
\begin{aligned}
&sim(c_i, Q) = \sum_{t_k \in (c_i \cap Q)} \frac{1 + \ln(tf(t_k, c_i) + 1)}{(1-s) + s\dfrac{dl}{avdl}} \times tf(t_k, Q) \times Icf(t_k) \\
&f(t_k, c_i) = balance\_weight(t_k) = DF(t_k) * Aver\_RankValue(t_k) \\
&Icf(t_k) = \ln(\frac{N+1}{n_{t_k}})
\end{aligned}
\tag{5}
$$

Where $tf(t_k,c_i)$ is the frequency of the t_k in the *ith* cluster, $tf(t_k,Q)$ refers to the frequency of the t_k in query Q. $Icf(t_k)$ is the inverse cluster frequency.

4 Ranking Features Selection

After the candidate clusters are determined, we need choose the feedback fragments of high quality from them. Therefore, some features are introduced into the paper to assist the related fragments selection and meanwhile, the low quality fragments are filter out.

(1)Relevance Score(R_Score): In the information Retrieval, the use of the relevance score feature implies that the higher a document is ranked in the first-pass retrieval, the more chance it can be a good feedback document. In this paper, following equation is used to measure it.

$$
sim(p_i, Q) = \sum_{t_k \in Q} \frac{1 + \ln(tf(t_k, p_i) + 1)}{(1-s) + s\dfrac{dl}{avdl}} \times tf(t_k, Q) \times (1 + \log\frac{N}{N_k}) \tag{6}
$$

(2)Co-occurrence of the query terms(Co_Weight): we think that if queries is often co-occcurrence in the same windows unit of one document, then the document is relevant to the query. The higher frequency of term co-occurrence , the more relevant between the document and query intention is.

$$
Co\_Weight(p_i) = \frac{1}{P\_TNum} * \sum_{(t_i, t_j) \in Q} \frac{\sum_{s_k \in p_i, k=1}^{m} c\_tf(t_i, t_j, s_k)}{tf(t_i, p_i) + tf(t_j, p_i)} \tag{7}
$$

Where *P_TNum* is the number of query terms, $c\_tf(t_i,t_j,s_k)$ refers to the frequency of word pair(t_i,t_j) co-occurrence in the windows unit s_k. In this paper, s_k. is defined as the leaf node or mixture node in XML document, m denotes the number of leaf node or mixture node in p_i. $tf(t_i,p_i)$ and $tf(t_j,p_i)$ are the frequency of the term t_i and t_j in fragment pi respectively.

(3) Entropy: The Entropy feature measures how the query terms are spread over a given feedback document. The PRF process extracts the most informative terms from the feedback documents. In many cases, there might be only a small part of the feedback document that contains relevant information. Thus, off-topic terms are possibly added to the query, resulting in a decrease in the retrieval performance. Therefore, it is necessary to examine the distribution of query terms in the feedback documents to see to which degree the feedback documents are related to the topic. In our work, Entropy is defined as follows:

$$Entropy(Q, p_i) = \sum_{t \in Q} (-\sum_{k=1}^{m} (p(t, s_k) * \log_2 p(t, s_k))) \tag{8}$$

where $p(t,s_k)$is the probability of observing the query term t in the *kth* subset of the fragment in tokens. In this paper, we empirically set leaf node or mixture node as one subset in a fragment. In order to avoid assigning zero probability to subsets where the query term does not appear, we apply Laplace smoothing as follows:

$$p(t, s_k) = \frac{tf(t, s_k) + 1}{tf(t, p_i) + m} \tag{9}$$

where $tf(t,s_k)$ is the term frequency in the *kth* subset of the fragment, and $tf(t,p_i)$ is the term frequency in the whole fragment. *m* is the number of subsets that the fragment is divided into.

(4) Cluster_Sim(p,C): The similarity between a given feedback fragemnt *p* and the candidate cluster C. In this paper, *Sim(p,C)* is measured indirectly through the similarity between the candidate fragments and the cluster center.

(5) Cluster_RValue: The Rank Value of the candidate clusters. The candidate fragments lies in the candidate clusters. If the rank value of the cluster is front, it shows that the relevance is very high and hence the probability that the candidate fragment becomes good feedback fragment is much larger.

In summary, we define the evaluation formula of feedback fragments as follows:

$$\begin{aligned} Final\_Sim(d_i, Q) = Cluster\_RValue_i * \alpha * Co\_Weight(p_i) * R\_Score(p_i, Q) \\ + \beta * Cluster\_Sim(p_i, c_i) + \gamma * Entropy(Q, p_i)) \\ \alpha + \beta + \gamma = 1 \\ Cluster\_RValue_i = (ClusterNum - R_i) \big/ ClusterNum \end{aligned} \tag{10}$$

Where *ClusterNum* refers to the total number of the clusters, R_i is the rank value of the cluster which the fragment pi belongs to.

5 Experiment Analysis and Evaluation

In the following, we perform the experiments for good pseudo relevance feedback fragments selection. We use indri[8] for both indexing and retrieval. The data collection adopts IEEE CS, a text-centric dataset provided by INEX 2005, which provides the evaluation results according to 29 different queries. and reach to totally 170001.

An initial step of our experiments is to perform search results clustering. Details of results could be seen in [7]. In this section, our work focus on the good xml fragments selection based on k-mediod clustering results.

5.1 Clusters Ranking Model Based on Cluster Labeling Evaluation

Users usually think that the more relevant fragments, the more top position be ranked. So, our ultimate target not only find good feedback fragments but rank in the front position as soon as possible. To reach this goal, the first premise is that the candidate relevant clusters should be identified correctly. So, our baseline is cluster center fragment's ranking model, in which the similarity computation is performed between the real cluster center fragment and query. Table I provides the comparison results, in which the candidate clusters are selected through either cluster labeling based or cluster center fragment based. In this table, according to the INEX 2005 relevance evaluation results, the relevant fragment distribution is expressed as the form $C_i(R\_PN_i/PN_i)$. C_i is the *ith* cluster, R_PN_i is the number of relevant fragments in the *ith* cluster, and the PN_i refers to the total number of fragments in the *ith* cluster. The symbol of *#RNum* stands for the number of relevant fragments and the value is equal to the sum of the number of relevant fragments in the candidate clusters. It is obviously that if one method gets the bigger value of *#RNum* than others, it proves to be better than others.

Table 1 shows encouraging results. Compared to the clusters ranking model based on cluster center fragment, our proposed method displays better performance and obtains the much righter candidate clusters. It illustrate fully from the side that the cluster labeling extraction based on equalization weight get proper cluster labeling and lead to follow-up right selection of candidate clusters. On the contrary, in the ranking model based on cluster center fragment, the candidate clusters are chosen completely by the similarity between the real cluster center fragment and query. As matter of fact, the real cluster center fragment sometimes does not necessarily reflect the main idea of the whole cluster although the similarity is high. Therefore, the method based on cluster center fragment is easy to bring to the noise and cause to poor results.

Table 1. The candidate relevant clusters results

Query No.	Relevant clusters	Cluster Labeling		Cluster Center Fragment	
		Candidate Clusters	#RNum	Candidate clusters	#RNum
202	1(1/68),10(2/55)	1,11	1	14,9	0
203	1(1/12),3(14/79),5(1/54),8(2/11)	3,5	15	3,1	15
205	1(3/34),2(13/120)	2,4	13	1,3	3
206	1(13/159),7(1/3),13(1/2)	3,1	14	3,1	14
207	2(32/96),5(1/3),6(1/4),8(1/4),12(12/33), 13(1/12)	2,12	44	2,12	44
208	1(28/136),3(1/30)	1,3	29	1,3	29
209	2(79/143),3(12/29),4(3/25)	2,3	91	2,4	82
210	1(16/123),2(1/18),5(4/18),13(3/12)	1,2	17	2,13	4
212	1(1/20),2(18/71),4(15/70),6(4/19)	2,5	18	5,6	4
213	4(3/85),11(9/52),13(2/13)	4,7	3	4,14	3
216	1(4/67),4(3/32),5(1/8),9(11/72)	1,9	15	4,1	7
217	1(1/119)	1,8	1	1,10	1
218	1(8/78),4(17/46),6(1/9),9(1/10),10(2/4), 11(3/11),12(1/15)	4,12	18	3,7	0
219	2(7/135)	2,6	7	10,2	7
221	1(18/132),4(2/11)	1,4	20	2,5	0
222	1(6/22),2/19),3(2/12),4(10/36),5(2/8), 7(1/6),8(1/16),9(18/48),10(1/7),12(1/15), 13(1/5)	4,9	28	7,4	11
223	1(14/59),3(32/96),5(2/12),6(1/13)	3,1	46	1,3	46
227	2(19/115)	2,4	19	4,3	0
228	1(27/110),6(3/11),9(9/40),10(4/10), 13(1/4),14(1/4)	1,9	36	10,1	31
229	1(11/121),2(3/22),3(1/7),4(1/6)	1,2	14	10,9	0
230	1(7/124),5(1/5),8(2/11)	1,14	7	4,13	0
232	2(15/131),7(1/2),8(1/5),9(1/5),11(1/23), 12(2/7)	2,11	16	7,5	1
233	1(5/148),6(1/2),8(2/4)	1,5	5	1,2	5
234	1(3/45),2(4/16),4(6/28),5(8/18),6(4/12), 7(1/9),8(4/10),9(8/11),11(23/37),12(1/3)	1,4	9	7,1	4
235	1(9/54),2(6/38),3(11/42),5(1/10),7(1/28), 9(10/12)	3,9	21	5,2	7
236	1(27/54),2(8/10),5(8/24),6(3/88)	1,6	30	4,5	8
237	1(30/151),2(1/2),3(1/8),7(3/4),8(1/27)	1,8	31	8,1	31
239	3(2/7),4(7/30),7(14/18),9(7/87),12(1/6), 13(1/6),14(2/9)	4,9	14	13,12	2
241	7(6/117)	7,13	6	4,5	0

5.2 Fragments Ranking Model Evaluation

After the candidate clusters are selected, the fragments in them are ranked in descent. In the equation(10), three different parameters values may result in different performance. So, We firstly optimize the value and the value of $\alpha = 0.2, \beta = 0.7, \gamma = 0.1$, leads to best performance.

In order to verify the effectiveness of the proposed ranking features, based on above the optimal parameters value, we compared the performance on precision between original query results and our approach. Table 2 provide the experimental results on top 5,10,15 and 20 fragments respectively. In the tables, each cell in the last four columns contains the obtained AP value, and a star indicates a significant improvement.

Table 2. Precision Comparison of Feedback Fragments

Method	AP@5	AP@10	AP@15	AP@20
Original retrieval results	0.32	0.34	0.33	0.31
Clustering and Ranking Model Results	0.48 (*0.5)	0.42 (*0.24)	0.39 (*0.15)	0.34 (*0.10)

From the data of the results, we can draw the following conclusion. On the one hand, compared to the original results, the proposed fragment ranking model could return much more number of good feedback fragments and meanwhile most of them are top position. The performance has been improved by 50%, 24%,15% and 10% on the top 5,10,15 and 20 fragments respectively, which further validate from the side that the candidate cluster selection is rationality and validity. On the other hand, the proposed ranking features could express user's query intention and make the related fragments return in the top position as soon as possible.

6 Conclusion

Ensuring high quality of pseudo relevance feedback document set is the first problem of avoiding query drift in PRF. In this paper, we have studies how to identify and select good feedback fragments and presents a complete approach. Based on search results clustering, cluster labeling extraction is firstly performed, and the candidate relevant clusters are identified by using cluster key terms. Subsequently, several ranking features are introduced to help relevant fragments identification from the candidate relevant clusters. The experimental results show that the proposed method is reasonable and could obtain high quality XML pseudo relevance feedback fragments set.

Acknowledgements. In this paper, the research was sponsored by the Social Science Foundation of China (Project No.12CTQ042) and Jiangxi Department of Education Science and Technology Project (Project No. GJJ11729).

References

1. Kyung, S.L., Croft, W.B., James, A.: A Cluster-Based Resampling Method for Pseudo-Relevance Feedback. In: Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 235–242. ACM Press, New York (2008)
2. Ben, H., Ladh, O.: Finding Good Feedback Documents. In: Proc. of the 18th ACM Conf. on Information and Knowledge Management (CIKM), pp. 2011–2014. ACM Press, New York (2009)
3. Raman, K., Udupa, R., Bhattacharya, P., Bhole, A.: On Improving Pseudo-Relevance Feedback Using Pseudo-Irrelevant Documents. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 573–576. Springer, Heidelberg (2010)
4. Sakai, T., Manabe, T., Koyama, M.: Flexible Pseudo-Relevance Feedback via Selective Sampling. ACM Transactions on Asian Language Information Processing 4(2), 111–135 (2005)
5. Shariq, B., Andreas, B.: Improving Retrievability of Patents with Cluster-Based Pseudo-Relevance Feedback Document Selection. In: Proc. of the 18th ACM Conf. on Information and Knowledge Management (CIKM), pp. 1863–1866. ACM Press, New York (2009)
6. Kevyn, C.T., Jamie, C.: Estimation and Use of Uncertainty in Pseudo-Relevance Feedback. In: Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 303–310. ACM Press, New York (2007)
7. Zhong, M.: Combining Term Semantics with Content and Structure Semantics for XML Element Search Results Clustering. Journal of Convergence Information Technology 7(15), 26–35 (2012)
8. Carnegie Mellon University and the University of Massachusetts. INDRI: Language Modeling Meets Inference Networks (March 2010), http://www.lemurproject.org/indri/

Informed Weighted Random Projection for Dimension Reduction

Jaydeep Sen and Harish Karnick

Indian Institute of Technology, Kanpur,
U.P:208016, India
jaydeepsen18@gmail.com, hk@cse.iitk.ac.in

Abstract. Dimensionality reduction is a frequent pre-processing step in classification tasks. It helps to improve the accuracy of classification by better representing the dataset and also alleviates the curse of dimensionality by reducing the number of dimensions. Traditional dimensionality reduction techniques such as PCA or Kernel PCA are well known techniques that find a lower dimensional subspace which best represents the higher dimensional dataset. On the other hand, random projection can also be considered as a dimension reduction technique that tries to approximate the same topology of higher dimensional data in a lower dimensional space. Both approaches reduce dimensions but because of their different objectives they have not been successfully integrated. Here we show that in practice and more specifically in a supervised setting like classification, we can link the two methods to make random projection more informed in making the low dimensional representation competitive with the original data set with respect to classification accuracy. In this paper we propose a novel dimensionality reduction technique, namely informed weighted random projection, that combines Kernel PCA and random projection in an efficient way. The kernel PCA algorithm is applied initially to obtain a sub-space of reduced dimensions then the new lower dimensional bases derived by the kernel PCA are weighted in proportion to the measured robustness coefficient of each base. The proposed dimensionality reduction scheme has been applied on several benchmark datasets from the UCI repository and experimental results show that informed weighted random projection attains higher accuracy than the usual unweighted combination for all the datasets used in our experiments.

Keywords: Dimension Reduction, Kernel PCA, Random Projection, Classification.

1 Introduction

The time and space complexity of classification algorithms often grow exponentially with the size of the input vector [1]. Moreover, the number of samples required to train a classifier grows exponentially with the dimension of the feature space and is commonly known as the curse of dimensionality [1] [2] problem.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 433–442, 2013.

Thus, inevitably, dimensionality reduction becomes important. The goal is to generate a reduced set of features that is still sufficient to represent the data and preserve those properties that help in tasks like classification. Dimension reduction has basically two aims: first to shrink the original dimension of the feature vector to a reasonable size and second to improve the classification accuracy by retaining the most discriminatory information and removing the irrelevant and redundant information.

We propose a new approach to reduce the dimension in two steps. In the first step, we use principal component analysis (PCA) [3] [4] to obtain a linear combination of high dimensional features that generate the lower dimensional data with uncorrelated dimensions [3] having maximum variance [3]. So PCA acts as a filter to represent the original dataset by a linear subspace of the most significant dimensions while removing correlated, redundant ones. In the second step we further reduce the dimension through random projection [5] by treating the PCA output as the target topology to be preserved. Random projection derives its benefit from the Johnson-Lindenstrauss lemma [6] which shows that we can construct a lower dimensional representation of a higher dimensional dataset, where with high probability, the expected distance between any pair of points in the lower dimensional representation is approximately the same as in the higher dimensional space [6]. It is in this step that we use the label information to make the random projection better informed. Instead of just using PCA output as the target topology, we create a more robust representation of the PCA output by weighing PCA bases with different weights so as to maximise the *robustness of the representation* (explained in the following section) and use it in the random projection to generate a better lower dimensional representation. We show through experiments that this information actually helps to retain the original accuracy with even fewer dimensions in most cases while the simple integration of PCA and Random Projection fails. Thus efffectively the integration process is optimized by preserving a weighted version of PCA output. Because each new feature in the PCA reduced lower dimensional space is weighted in proportion to the robustness coefficient of that feature it makes the resultant distance metric between the two data points more robust and hence particularly useful for classification.

The rest of the paper is organized as follows. Section 2 discusses PCA and Random Projection as dimensionality reduction schemes. Section 3 describes the integration of the two processes to yield a more effective and possibly optimum reduction method. Section 4 describes the mathematical formulation and the algorithm. Section 5 gives the results on standard UCI datasets and section 6 concludes the paper with possible pointers for future work.

2 Background

Usually, the input data often inherently contains some patterns or dependency structures. These are often hidden within the irrelevant features and adversely affect performance. The aim of a good dimension reduction scheme is to unveil

the hidden correlations and construct a new subset of features that can best represent the properties of the data and help in building an accurate classifier. For classification it is important to realize that some of these features may have different discriminative power and removing them results in loss of information. Hence, dimensionality reduction is implemented as an optimization procedure to search for an optimal subset of features that satisfies a desired measure [7]. In the next section we first present the kernel PCA method and using it as a base develop the informed weighted random projection scheme.

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a standard technique in statistical data analysis widely used in feature extraction and data compression. The goal of PCA is to reduce dimensionality of data while retaining the maximum variation present in the original dataset. In PCA, the redundancy is measured by correlations between data elements and so the uncorrelated features represent the lower dimensional subspace. PCA [3] [4] [8] performs dimensionality reduction by embedding the data into a linear subspace with fewer dimensions.

PCA produces a linear subspace of the original higher dimensional feature space where each pair of dimensions in the PCA output is uncorrelated and so not redundant. Given the original feature space X (say), we obtain the PCA output Y(say) as the reduced dataset such that

$$Y = XM, \tag{1}$$

where M is constructed using eigenvectors of $cov(X)$. $cov(X)$ is defined as the covariance matrix [8] obtained using the input data set X. Kernel PCA is an extension of traditional PCA to handle non linearity. Kernel PCA works in the same way as PCA but with the input dataset transformed using a suitable kernel function which can be Gaussian or polynomial or linear.

2.2 Random Projection

In random projection dimension reduction features are not retained or discarded by measuring their importance unlike traditional feature reduction algorithms. Instead the method tries to preserve approximately the same high dimensional topology but in a lower dimensional representation with a probabilistic measure that guarantees the approximate preservation of distance. It is actually a randomised computational process to construct a lower dimensional representation from a higher dimensional one. The dimensionality reduction obtained by random projection is oblivious of the higher dimensional patterns or details. It is a purely mathematical step that does not depend on the orientation of higher dimensional points at all. Suppose we have input data in a matrix $A_{n\times d}$ of n data points in d dimensional space, then by the Johnson and Lindenstrauss lemma (1984) [6] there exists a randomized construction $R_{d\times k}$ such that if we define the projection of A to be

$$E = AR \tag{2}$$

then the pairwise distance between any two points of A in the higher dimensional space is approximately preserved in the projected space E. The exact construction of matrix R can be found in the paper by Arriaga and Vempala [9]. Arriaga and Vempala showed that we can construct the random matrix $R_{d\times k}$ such that each element of R_{ij} is a random value taken from $N(0,1)$, a normal distribution with mean 0 and variance 1. The detailed proof is more general and shows that this actually holds for any distribution D which is symmetric with respect to the origin and

$$E(r^2) = 1$$

and even moments are bounded by a positive number [9]. With this construction of R, the precise mapping f from the d dimensional space A to the k dimensional space E can be written as follows

$$f(u) = \frac{1}{\sqrt{k}}(uR). \tag{3}$$

Furthermore, the accuracy measure of the mapping f is shown to be as follows, for a given ε, $k > 12 * \frac{\log n}{\varepsilon^2}$ and if $u, v \in A$, the following inequality holds.

$$(1-\varepsilon) \parallel u-v \parallel^2 < \parallel f(u)-f(v) \parallel^2 < (1+\varepsilon) \parallel u-v \parallel^2 \tag{4}$$

3 Integrating Kernel PCA and Random Projection

Kernel PCA and random projection methods are integrated using the lemma proved by Arriaga and Vempala [9], stated below.

Lemma 1. *An l robust half space (ε, δ) is learnt by projecting a set of m points to R^k where, $k = \frac{100}{l^2} \ln \frac{100}{\varepsilon l \delta}$ and $m = O(\frac{1}{l^2}\frac{1}{\varepsilon} \ln \frac{1}{\varepsilon l \delta})$*

The robustness value l is defined as the minimum separation distance between two examples belonging to different classes [9]. The key observation in the above lemma is that k and m are inversely proportional to the robustness value l. Hence for a fixed number of examples say, m, in the higher dimensional space the value of l is the only critical handle that we have to control and reduce the value of k. Once we fix k then for a fixed value of m and k the accuracy ε is inversely related with l^2. In the integration part we specifically exploit this relation by increasing the value of l which in turn improves accuracy for a fixed value of m and k. We do this by weighting different dimensions of the kernel PCA reduced dataset with different weights so chosen that for each dimension the weight assigned is a measure of how robust that dimension is in the kernel PCA reduced dataset reflecting the contributions of that dimension in the total robustness value l in the reduced dataset. We call the weights the robustness factor of the dimension. Initially we define it as follows:

For two classes $C1$ and $C2$ having $\mid C1 \mid$ and $\mid C2 \mid$ number of examples, the robustness factor for dimension i is computed using equation (5)

$$(r.f)^i = \frac{\mid \sum_{a \in C1} a^{(i)} - \sum_{b \in C2} b^{(i)} \mid}{sdv^{(i)}} \tag{5}$$

a, b are sample points in the training data set and a^i, b^i denote the value of the i^{th} dimension of these sample points respectively. $sdv^{(i)}$ is the standard deviation of the i^{th} dimensional value for the whole datatset i.e $C = C1 \bigcup C2$. The intuition is if we have information like class labels we can rank significant dimensions according to their discriminative power which we measure as the mean difference across the classes. If the mean difference between points from different classes has a high value for a particular dimension then it will have a high numerator value in equation (5) and hence a high robustness factor indicating it is a significant contributor in the total robustness value in the set C. But the mean difference alone cannot be a measure to compare different dimensions because of differences in the variance among the dimensions. A dimension which has high variance is likely to have high mean difference too because of its greater span of values. So, to make the robustness factor comparable among different dimensions, the mean difference is scaled by the standard deviation of that dimension. Although the computation of the robustness factor in equation (5) is measured over the whole dataset C one must note that robustness is originally defined only on the two nearest points from separate classes. Hence, instead of considering the whole dataset to compute the robustness factor, in practice we need only consider points within a certain min-length strip. We fix this min-length to be double the distance between the nearest two points of different classes. So, we first find the nearest points of opposite classes and fix the min-length to be double of that value and use different class examples within that min-length to compute the robustness factor as follows

$$(r.f)^i = \frac{\sum_{(a,b)\in S} \mid a^{(i)} - b^{(i)} \mid}{\mid S \mid sdv^i} \tag{6}$$

where, $S = \{(a, b) \mid a, b \in C, label(a) \neq label(b), \parallel a - b \parallel < min - length\}$ i.e. S is the set of pairs of points (a, b) where a and b are from different classes and the distance between a and b is less than min-length. We calculate the robustness factor of each dimension over this set S.

The measure of robustness factors reflects the fact that the higher is the mean difference for a dimension the more is its weight. The corresponding weight factors are chosen in proportion to their robustness factors and normalized to sum up to 1. We construct a matrix W consisting of weight factors of different dimensions and use it to get the weighted Kernel PCA reduced data set

$$Z = YW. \tag{7}$$

Since the dimensions are uncorrelated [7] and chosen according to the highest variance [7], the weight factors rank the dimensions according to their contributions to the robustness values. If we consider data set Z, intuitively it is plausible that Z will have higher robustness value than Y and so the random projection performed on Z will yield better results than on Y. This is the core idea of the proposed weighted random projection approach which we support by experimental verification.

Algorithm: Weighted Random Projection. The higher dimensional data set is preprocessed before applying the proposed algorithm. The preprocessing step scales the data to ensure all the dimensions span the range $[-1, 1]$. Fixing the range through scaling ensures the weights assigned to different dimensions are as per their importance in the classification task and do not depend on the range they cover. Without scaling, a dimension that spans a larger range may have higher weight than one spanning a lower range but more effective in contributing to the robustness value and hence classification. So, in effect scaling homogenizes all the features before computing weights. Since the lemma connecting robustness to accuracy in section 3 is primarily established considering examples in a unit ball, so we pre-process the data so that they lie in a unit ball, and thus exactly satisfy the condition of the lemma. This pre-processed data set is used in the weighted random projection algorithm.

Algorithm: Weighted Random Projection
Input: High dimensional data set: $X_{(m\times n)}$
Output: Lower dimensional representation: $WeightedRandomReduced_{(m\times k)}$

1. Apply KernelPCA to $X_{(m\times n)}$ to obtain $Y_{(m\times d)}$.
2. Compute $(r.f)^i$ using equation (6).
3. Normalize $(r.f)^i$ to obtain $(w.f)^i$ by $(w.f)^i = \frac{(r.f)^i}{\sum_i (r.f)^i}$.
4. Scale $Y_{(m\times d)}$ with weight matrix $W_{(d\times 1)}$ to obtain $Z_{(m\times d)}$, where $W = (w.f)^{i^T}$.
5. Perform Random Projection on $Z_{m\times d}$.
   Repeat 1000 times
   5.1 Construct $R_{(d\times k)}$, each element chosen randomly from $N(0, 1)$.
   5.2 $M = M + Z.R$.
   5.3 $M = M/100$ to have unit length.
   5.4 $M = M/\sqrt{k}$.
6. Output: $WeightedRandomReduced_{(m\times k)} = M$.

4 Experiments and Results

We tested our weighted informed random projection algorithm on benchmark datatsets from the UCI repository. The datasets chosen for the experiment were (1) spambase: [10], (2) Ionosphere: [11], (3) WDBC: Wisconsin Data for Diagnosis of Breast cancer [12], (4) Heart: Data for Heart disease diagnosis [13], (5) Diabetes: Data for Diabetes disease diagnosis[14], (6) WPBC: Wisconsin Data for Breast Cancer prognosis [15], (7) Wine: [16], (8) ILPD: data for Indian Liver Patient Diagnosis[17], (9) Australlian Credit approval data [18], (10) Liver Disorder Dataset[19], (11) Glass dataset for classification[20]. We emphasize that the informed weighted random projection is not about designing a new classifier, so the effectiveness of the approach should not be measured only by comparing

it with the original dataset accuracy. We must contrast the utility of weighted random projection against simple reduced projection and show that it gives better accuracy than other representations in a space of the same dimension for any classifier. So our results compare the weighted reduced representation against same dimensional different representations and not just with the original dataset which has a different dimension. In the paper by Vempala et. al. that established the lemma connecting robustness with accuracy the algorithm was a half space algorithm so we choose the Perceptron classifier (close to half space algorithm) and the SVM classifier which is very widely used in practice to test our scheme.

We have experimented with both variants of the weighting scheme by considering mean difference alone and by considering mean difference scaled by standard deviation. Robustness factors are calculated considering both the whole dataset and across only a min-length strip. The reduced dataset dimensions are three fourths of the original dimension in KPCA reduction and then half the original dimension after random projection. The naming convention for different weighting schemes are as follows:

W1: uses mean difference as weight and robustness factor is computed across the whole dataset.

W2: uses mean difference scaled by standard deviation as weight and robustness factor is computed across the whole dataset.

WL1: uses mean difference as weight and robustness factor is computed across only the min-length strip.

WL2: uses mean difference scaled by standard deviation as weight and robustness factor is computed across only the min-length.

The datasets we generate through different weighted schemes and reductions are named as follows:

The columns in the tables are named as: *Org* is for the original dataset; *kpca* is for kernel PCA reduced dataset; *kpcaW1*, *kpcaW2*, *kpcaWL1*, *kpcaWL2* are for weighted representations of Kernel PCA reduced dataset using weights W1, W2, WL1, WL2 respectively;. *rkpca* means twice reduced dataset using random projection after kernel PCA reduction while *rkpcaW1*, *rkpcaW2*, *rkpcaWL1*, *rkpcaWL2* are reduced by random projection from datasets *kpcaW1*, *kpcacW2*, *kpcaWL1*, *kpcaWL2* respectively. Table 1 below shows robustness measures for different datasets with different weighted representations.

Table 1. Robustness value for each dataset with different weighting schemes

Dataset	org	kpca	kpcaW1	kpcaW2	kpcaWL1	kpcaWL2	rkpca	rkpcaW1	rkpcaW2	rkpcaWL1	rkpcaWL2
spam	0.025	0.030	0.035	0.030	0.035	**0.039**	0.006	0.007	0.009	0.008	**0.010**
Ionosphere	0.651	0.407	0.480	0.480	0.464	**0.516**	0.032	**0.044**	0.038	**0.044**	0.038
WDBC	0.074	0.842	1.079	1.079	**1.132**	1.227	0.156	0.181	0.198	0.183	**0.200**
Heart	0.039	0.253	0.128	0.128	0.316	**0.321**	0.020	0.015	0.023	0.015	**0.042**
Diabetes	0.034	0.084	0.089	0.092	**0.137**	**0.137**	0.043	0.067	0.067	**0.070**	**0.070**
WPBC	0.052	0.523	0.551	0.551	0.680	**0.714**	0.024	0.024	0.024	**0.028**	0.024
Wine	0.171	1.103	1.742	1.752	**1.809**	1.800	0.160	0.269	0.260	**0.272**	0.262
ILPD	0.292	0.003	0.005	0.005	0.003	**0.017**	0.001	0.002	**0.004**	**0.004**	**0.004**
Liver	0.108	0.050	0.031	0.051	0.066	**0.153**	0.003	0.002	**0.004**	0.003	**0.004**
Credit	0.507	0.197	0.148	0.148	0.212	**0.318**	0.005	0.005	0.005	0.007	**0.009**
Glass	0.123	0.024	0.023	**0.025**	**0.025**	0.024	0.004	0.005	0.005	**0.006**	0.005

A careful analysis reveals that for the same dimension weighted representations actually have higher robustness values. This specifically helps the classification task because a higher robustness value ensures higher accuracy for a half space algorithm as was shown by Vempala. So next we report accuracy results obtained using classifiers. Table 2 below gives the results obtained using SVM as a classifier with original and different weighted combinations of their reduced representations.

Table 2. Classification accuracy for each dataset with a SVM classifier

Dataset	org	kpca	kpcaW1	kpcaW2	kpcaWL1	kpcaWL2	rkpca	rkpcaW1	rkpcaW2	rkpcaWL1	rkpcaWL2
spam	60.54	60.68	62.55	60.74	61.61	**64.46**	60.59	60.61	68.81	60.63	**80.74**
Ionosphere	93.16	96.01	96.86	96.86	96.01	**97.15**	94.87	**95.72**	95.44	**95.72**	95.44
WDBC	98.94	97.71	97.06	97.06	97.71	**98.24**	93.84	97.18	97.83	97.36	**98.01**
Heart	84.81	82.96	80.37	80.37	**83.33**	82.22	81.85	79.02	82.96	80.00	**83.33**
Diabetes	76.95	71.87	72.39	73.69	**73.82**	**73.82**	66.92	72.00	71.87	**72.52**	72.39
WPBC	81.31	80.31	80.30	80.30	**80.80**	**80.80**	69.69	74.74	74.74	**77.77**	72.72
Wine	99.43	97.19	97.75	97.75	**98.31**	**98.31**	91.57	**97.19**	96.06	**97.19**	96.62
ILPD	61.40	34.64	**69.98**	68.43	36.36	69.12	30.70	68.29	**69.12**	68.61	**69.12**
Liver	42.89	66.66	62.02	67.24	67.53	**68.11**	66.37	60.66	67.82	68.11	**68.21**
Credit	85.65	76.27	73.56	73.71	79.27	**80.00**	74.20	73.55	74.56	78.84	**79.69**
Glass	42.52	49.53	48.59	50.93	**52.33**	50.93	39.25	46.26	46.26	**47.19**	46.72

Next in Table 3 we also list accuracy results obtained with the voted perceptron algorithm which is much like the half space algorithm used by Vempala to prove the lemma.

Table 3. Classification accuracy for each dataset with a voted perceptron classifier

Dataset	org	kpca	kpcaW1	kpcaW2	kpcaWL1	kpcaWL2	rkpca	rkpcaW1	rkpcaW2	rkpcaWL1	rkpcaWL2
spam	49.47	80.33	81.84	80.94	82.10	**84.64**	78.35	78.42	79.72	79.77	**81.79**
Ionosphere	85.50	90.50	92.91	93.02	92.42	**93.08**	86.93	**87.98**	86.99	87.93	87.15
WDBC	89.67	94.71	96.19	96.22	94.85	**96.77**	82.16	93.24	93.45	93.41	**93.66**
Heart	63.93	80.07	79.63	79.63	80.74	**81.93**	77.26	76.37	77.89	77.56	**78.44**
Diabetes	66.03	71.38	71.60	71.60	**71.96**	71.92	69.56	70.97	71.30	**71.88**	71.61
WPBC	76.29	76.24	**76.29**	**76.29**	**76.29**	**76.29**	74.20	74.38	74.20	**74.71**	74.20
ILPD	71.25	70.31	70.03	70.29	70.56	**71.32**	70.72	70.98	71.08	**71.15**	**71.15**
Liver	63.53	58.65	58.68	59.23	58.68	**59.29**	59.72	58.85	59.60	60.06	**61.20**
Credit	64.22	66.03	64.93	65.77	66.19	**67.17**	63.93	63.88	64.68	64.71	**65.72**

In both Table 2 and Table 3 the important observation is that we have very high correlation with table 1 i.e. a higher robustness value in table 1 helps attain higher accuracy with SVM and Perceptron classifiers in Table 2 and 3 respectively. This is particularly important because in the same dimensional space it demonstrates the utility of weighting representations over simple reductions given two benchmark classifiers. We further extract the best results that in effect summarize the effectiveness of the weighted scheme. Table 4 lists best accuracy values for the weighted representations along with the robustness measure which clearly reveals high correlation between the obtained accuracy and the robustness value in the weighted representation.

Table 4. Classification accuracy with SVM classifier along with robustness value for each dataset

Dataset	org		kpca		Best Weighted kpca		rkpca		Best Weighted Reduced kpca	
spambase	60.54	0.025	60.68	0.030	64.46	0.039	60.59	0.006	80.74	0.010
Ionosphere	93.16	0.651	96.01	0.407	97.15	0.516	94.87	0.032	95.72	0.044
WDBC	98.94	0.074	97.71	0.842	98.24	1.227	93.84	0.156	98.01	0.200
Heart	84.81	0.039	82.96	0.253	83.33	0.316	81.85	0.020	83.33	0.042
Diabetes	76.95	0.034	71.87	0.084	73.82	0.137	66.92	0.043	72.52	0.070
WPBC	81.31	0.052	80.31	0.523	80.80	0.714	69.69	0.024	77.77	0.028
Wine	99.43	0.171	97.19	1.103	98.31	1.809	91.57	0.160	97.19	0.270
ILPD	61.40	0.292	34.64	0.003	69.12	0.017	30.70	0.001	69.12	0.004
Liver	42.89	0.108	66.66	0.050	68.11	0.153	66.37	0.003	68.21	0.004
Credit	85.65	0.507	76.27	0.197	80.00	0.318	74.20	0.005	79.69	0.009
Glass	42.52	0.123	49.53	0.024	52.33	0.025	39.25	0.004	47.19	0.006

5 Conclusion

Both kernel PCA and random projection are separately established methods for dimensionality reduction. We show that in a supervised environment, specifically if we have class labels available, we can exploit a link between accuracy and robustness to efficiently integrate these two techniques to design a new dimension reduction scheme, namely informed weighted random projection. The method is a two step dimension reduction process. We have tested eleven UCI datasets with our dimension reduction scheme and show that it performs at the same level or better than the original dataset for classification accuracy but with far fewer dimensions. Experimental results support our claim that our approach is better than a simple two step reduction when measured in the same dimension space (with two classifiers). Informed weighted random projection can be used to develop more robust representations with reduced dimensions and can help to improve classifier accuracy and performance.

References

1. Bellman, R.E.: Adaptive control processes - A guided tour. Princeton University Press, Princeton (1961)
2. Donoho, D.L.: High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality, `http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/Curses.pdf`
3. Shlens, J.: A tutorial on Principal Component Analysis, Systems Neurobiology Laboratory, Salk Institute for Biological Studies (2005)
4. Movellan, J.R.: Tutorial on Principal Component Analysis, `http://mplab.ucsd.edu/tutorials/pca.pdf`

5. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250. ACM, New York (2001)
6. Dasgupta, S., Gupta, A.: An elementary proof of the Johnson-Lindenstrauss lemma, http://charlotte.ucsd.edu/~dasgupta/papers/jl.pdf
7. Saul, L.K., Weinberger, K.Q., Ham, J.H., Sha, F., Lee, D.D.: Spectral methods for dimensionality reduction, Semisupervised Learning. MIT Press, Cambridge (2006)
8. Jolliffe, I.T.: Principal Component Analysis. Springer (2002)
9. Arriaga, R.I., Vempala, S.: An algorithmic theory of learning: Robust concepts and random projection. J. Mach. Learn. 63, 161–182 (2006)
10. Forman, G.: UCI Machine Learning Repository (1999), http://archive.ics.uci.edu/ml/datasets/Spambase
11. Sigillito, V.: UCI Machine Learning Repository (1999), http://archive.ics.uci.edu/ml/datasets/Ionosphere
12. Street, N.: UCI Machine Learning Repository (1999), http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnotstic%29
13. UCI Machine Learning Repository (1997), http://archive.ics.uci.edu/ml/datasets/Statlog+%28Heart%29
14. Sigillito, V.: UCI Machine Learning Repository (1990), http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes
15. Street, N.: UCI Machine Learning Repository (1995), http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Progonostic%29
16. Aeberhard, S.: UCI Machine Learning Repository (1991), http://archive.ics.uci.edu/ml/datasets/Wine
17. Bendi, V.R., Babu, M.S.P., Venkateswarlu, N.B.: UCI Machine Learning Repository (2012), http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+liver+Patient+Dataset%29
18. Quinlan: UCI Machine Learning Repository (1989), http://archive.ics.uci.edu/ml/datasets/Statlog+%28Australian+Credit+Approval%29
19. Forsyth, R.S.: UCI Machine Learning Repository (1990), http://archive.ics.uci.edu/ml/datasets/Liver+Disorders
20. German, B.: UCI Machine Learning Repository (1987), http://archive.ics.uci.edu/ml/datasets/Glass+Identification

Protocol Specification Inference Based on Keywords Identification

Yong Wang, Nan Zhang, Yan-mei Wu, and Bin-bin Su

School of Computer Science and Engineering
University of Electronic Science and Technology of China, 611731
Chengdu, China
cla@uestc.edu.cn

Abstract. Protocol reverse engineering is becoming important in analyzing unknown protocols. Unfortunately, many techniques often have some limitations for few priori information or the time-consuming problem. To address these issues, we propose a framework based on protocol finite state machine (FSM) construction, which can infer the protocol specifications without any priori information of protocols. To improve our framework's efficiency, we identify the keywords before the finite state construction. Our framework constructs two FSMs, one is $L-FSM$ (language FSM) and the other is $S-FSM$ (state FSM). $L-FSM$ is to illustrate the protocol languages. $S-FSM$ shows protocol sessions' state transitions. We evaluate our framework with both binary and text protocol. The ARP and the SMTP are the target protocols as inputs. The precision rate and the recall rate are used for evaluation criterias in our experiments. The ARP's precision and recall rate are both reached 100%. The SMTP's precision rate is 100% and recall rate is almost 98%.

Keywords: keywords Identification, protocol languages, finite state machine.

1 Introduction

Network protocols are sets of standards and rules for certain network communications. Each protocol has its own function to regulate users' behaviors on the Internet. As many protocols are closed and have no public specifications about them, they may be utilized to transfer unsafe data secretly. So the details of these unknown protocols are helpful in Internet traffic monitoring and management. Generally, the details of the unknown protocols analysis not only contains the protocol languages, but also the protocol state machine.

The technologies of analyzing the protocol specifications are multitudinous to list. For instance, Ploygot [1] uses dynamic analysis and bases on the way that analyze the transferring with the source codes of the program. These techniques are limited with the source codes of the programs. The ReverX [2] is another system which analyzes the unknown protocols with two finite state machines (FSM). If the protocol is complicated or the input messages are rather long, the construction of the FSM becomes difficult and time-consuming.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 443–454, 2013.

In this paper, we propose a framework which can infer an unknown protocol specification without any priori knowledge. Two FSMs are constructed. The $L-FSM$ can accept all the target protocol messages. The $S-FSM$ generalizes all the possible messages transitions within one session. For the efficiency, the keywords identification which uses Jaccard index as threshold is adopted to compress messages before constructing the $L-FSM$. We output the protocol languages and its state machine as the protocol specification.

Our main contributions are:

- A new framework is proposed which has a good scalability, inferring the unknown protocol language and protocol state without any priori information. The input traces could contain both text-based and binary-based protocols.
- A new keyword-based method is proposed to improve our framework more efficient, for it can filter out non-associated information and reduce the framework overhead in constructing the protocol language FSM.
- A new FSM-based method is adopted in our framework, which could analyze the unknown protocol language and show their state transitions.

The rest of the paper is organized as follows. Section II is dedicated to the related work. Section III shows the system architecture. Our algorithms and framework details are introduced in section IV. In section V, we carry out experiments to evaluate our framework. We draw brief conclusions in Section VI.

2 Related Work

We divide our discussion of related work into three areas, namely keyword identification, protocol language extraction, and protocol state inferring.

Keyword Identification. Keyword identification indicates messages boundaries and infers protocol languages for unknown network protocols. Marshall A.Beddoe presents Protocol Informatics(PI) [3] in 2004, which employs bioinformatics sequence alignment algorithm to reveal similarities between messages. RolePlayer [4] analyzes protocol structure focusing on status flag and field based on messages sequences, whose basic idea comes form ScripGen [5]. That system becomes ineffective or even powerless when the protocol has variable fields. Discoverer [6] achieves keyword identification by tokenization and initial clustering network traces. By putting sequences of n-gram in to Latent Dirichlet Allocation [7], their framework can identify the n-gram distribution in each word.

Protocol Language Extraction. A protocol can been seen as a formal language whose syntax rules are specified through a grammar. Wondracek et al. [8] propose an approach that works by dynamically monitoring the execution of the application, analyzing how the program processes the protocol messages that it receives. Autoformat [9] extracts the protocol language by building a protocol field tree and using the it to store the identified fields. ProDecoder [10] generates the protocol language by exploiting the semantics of protocol messages. It uses the Information Bottleneck [11] algorithm to infer the final protocol language.

Protocol State Inferring. Protocol state plays an important role in protocol specification. Trifilo et al. [12] build state machines by obtaining from the order of messages in the traces and the values of this relevant field. P. M. Comparetti et al. [8] generate the state machine by building an augmented prefix tree. Xiao et al. [13] get protocol state using Grammatical Inference (GI) [14]. Veritas [15] employs Probabilistic Protocol State Machine (P-PSM) [16] to describe protocol state transitions. Antunes et al. [2] build FSM from clustered sequences.

3 Preliminaries

3.1 Design Goals

The basic goal of our framework is inferring the protocol specification of an unknown protocol without any priori knowledge.

- The framework can infer unknown protocol languages precisely without any priori information, which can accept both text and binary messages as inputs.
- The framework should be efficient that we utilize the keywords identification to filter out the non-association information.
- Two FSMs should be constructed, one obtain unknown protocol languages which is named as $L-FSM$ and the other simulate protocol state transitions which is named as $S-FSM$. Both two FSMs should be generalized.

3.2 System Architecture

Our framework consists three components. Fig.1 is the architecture of our framework. The overviews of our framework are described as follow.

Keywords Identification. In this component, network traces of a specific protocol is collected carefully. Packets in these traces are assembled into the training messages. These messages are tokenized and the threshold TH is adopted to filter out tokens. Varying the value of TH, we can get the optimal tokens set $Tokens = \{T_1, ...T_i\}$. These optimal tokens are the keywords for text protocols. But for binary protocols, we splice the tokens to get keywords as long as possible.

Protocol Language Inferring. After we get the keywords set $K = \{K_1, ..., K_n\}$, the framework will construct a $FSMT$. The methodology is constructing the FSM using the keywords. We utilize the Moore reduction procedure for deterministic finite automata minimization [17] to generalize this FSM into a concise one. The generalized FSM will be treated as the protocol $L-FSM$.

State Machine Building. The purpose of this component is constructing a $S-FSM$ to simulate the state transition within one session. We utilize each format of the protocol languages as one transition to construct this $S-FSM$. Through the generalizing the S, we will get the $S-FSM$.

4 Our Framework

Our framework's input is the network traces of a specific protocol in the real world. Note that these messages of one protocol typically have multiple types, the keywords are identified before constructing the $L-FSM$. The $L-FSM$ is constructed using these messages' regular expressions.The $S-FSM$ is constructed in *State Machine Building*.

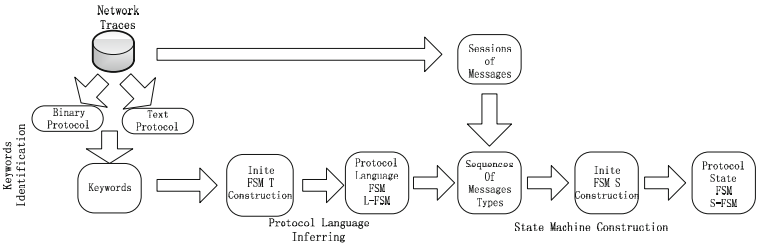


Fig. 1. The System Architecture

4.1 Keyword Identification

The messages of a specific protocol are input to this component. We propose a new method which is based on the keywords identification to infer the protocol language. The process is described in algorithm 1.

We firstly reassemble the input packets into messages. The next process that we called tokenization is breaking messages into tokens. The tokenization methods are different between text and binary protocols. In the text protocols, we identify test bytes by comparing them with the ASCII values. Then, we use the " $< SP >$ " and " $< TAB >$ " to represent token separators, and the " $< CRLF >$ " to represent message delimiter respectively. For binary protocols, we assume a single binary byte to be a binary token.

The Jaccard Index is used to filter these unfrequent tokens. We randomly divide messages set into two groups $\mathbb{A}$, $\mathbb{B}$ and each group contains all the types of the messages. After which, we store each token's frequency as $\mathbb{A} = \{T1_1 : F1_1, ..., T1_n : F1_n\}$ and $\mathbb{B} = \{T2_1 : F2_1, ..., T2_n : F2_n\}$. Where $T1_i$ is the i-th token and its frequency is $F1_i$ in group $\mathbb{A}$, $T2_i$ is the i-th token and its frequency is $F2_i$ in group $\mathbb{B}$. We calculate the similarity of $\mathbb{A}$ and $\mathbb{B}$ to filter the non-association tokens. We change a little about the Jaccard index's definition to adapt our apply. The new definition is as followed:

$$J(A,B) = \frac{\sum_{i=1}^{n} T1_i * T2_i}{\sum_{i=1}^{n} T1_i^2 + \sum_{i=1}^{n} T2_i^2 - \sum_{i=1}^{n} T1_i * T2_i}$$

In this expression, $\mathbb{A}$ and $\mathbb{B}$ are the two given groups that are transformed to vectors of frequent tokens. $T1_i$ represents the ith token in the group $\mathbb{A}$, and

$T2_i$ represents the *ith* token in the group $\mathbb{B}$. $J(\mathbb{A}, \mathbb{B})$ gains its maximum value 1 when all the tokens and their distributions in one group are the same with the other. It will achieve its minimum value 0 when all the items are distinct.

Algorithm 1. The Keywords Identification algorithm

Input: Messages Set $Messages = \{M_1, M_2, ..., M_n\}$ and the threshold set $Thresholds = \{TH_1, TH_2, ...TH_k\}$
Output: The keywords set $Keywords = \{K_1, K_2, ..., K_m\}$

```
Fre(T) ← 0, Tokens = {T1, T2, ...Th}, Token ← φ, Messages ← φ, Threshold ← φ
for each Mi ∈ Messages and Mi is text protocol do
    for each byte B ∈ Mi do
        for B ≠' \0' do
            for (B ≠ TAB) ∩ (B ≠ SPACE) do
                Token+ = B
            end for
            if Token ← UNDEFINED then
                t ← NewToken(), t ← Token, Fre(t) ← 1
            else
                t ← Token, Fre(t) + +
            end if
        end for
    end for
end for
for each Mi ∈ Messages and Mi is binary protocol do
    for each byte B ∈ Mi do
        Token = B
    end for
    if Token ← UNDEFINED then
        t ← NewToken(), t ← Token, Fre(t) ← 1
    else
        t ← Token, Fre(t) + +
    end if
end for
for each threshold THi in Thresholds do
    for each token Tj in Tokens do
        Filter the non-frequency Ti with the Threashold_i
    end for
    Scan the Jaccard set and find out the first maximum with THi
end for
if The token Ti is the text protocol then
    Keywords ← Tokens filtered by THi
else
    the Tokens Splicer with Tokens filtered by Threshold
end if
```

When we vary the value of $Threshold$ change, the groups' members T_i will change and so do the Jaccard index J. The first maximum J_j is what we need and the filter will filter the unfrequent tokens whose frequency is bellow the $Threshold_j$. After this phase, the keywords of text protocols messages are identified. But for binary protocols, we replay a group of messages and splice tokens as long as possible to identify keywords. The procedures of this splicer are described in Tokens Splicer.

4.2 Protocol Languages Extracting

The protocol language $FSM\ L - FSM = (Q, \sum, \delta, q_0, F)$ is defined as followed. Q is a finite, non-empty set of states. $\sum$ is the input alphabet which is composed

The Tokens Splicer

Step1: One message m_1 in ϕ_m are chose and replayed. We adopt the sequence alignment algorithm to location the tokens in this message.
Step2: If one token is followed by another, we splice them to get a candidate keyword. And the tokens are spliced as long as possible.
Step3: Another message m_2 in ϕ_m is chose. We operate on it just like step 2. If keywords obtained in step 3 is different with the corresponding one in m_2, we divide them into several ones.
Step4: Messages in ϕ_m are replayed and chose one by one. We align them from left to right by byte following step 4.
Step5: All the keywords identified above are collected, we get the keywords set ϕ_k.

of all the keywords. δ is the state-transition function while a new keyword is input: $\delta : Q \times \sum \rightarrow N$. q_0 is the initial state. F is the set of final states.

Construction of the Initial FSM. The input to this component is a set of keywords that from the alphabet of the protocol language. We construct an initial FSM to accept all types of the input messages. The details of $L-FSM$ construction are described in Algorithm 2.

Algorithm 2. The protocol Language Inferring

Input: messages regular expressions set $Messages = \{M_1, M_2, ..., M_n\}$,
the threshold $Threshold_f$, the alphabet $Alphabet = \{K_1, K_2, ...K_m\}$,
Output: The protocol language FSM $L = (Q, \sum, \delta, q_0, F)$
$q_0 \leftarrow NewState()$, $\sum \leftarrow Alphabet = \{K_1, K_2, ...K_m\}$,
$Q \leftarrow \phi$, $F \leftarrow \phi$, $\delta(q, k) \leftarrow UNDEFINED$
for $M_i \in Messages\{M_1, M_2, ..., M_n\}$ **do**
  $q \leftarrow m$
  for each $K_j \in M_i$ **do**
    if $\delta(q, K_j) \leftarrow UNDEFINE$ **then**
      $p \leftarrow NewState()$, $Q \leftarrow Q \cup \{p\}$, $\delta(q, K_j) \leftarrow p$, $q \leftarrow p$
    end if
    $F \leftarrow F \cup \{q\}$
  end for
end for
Set ω is the pointer, $\omega \leftarrow q_0$, Statistics all the $q\omega$'s frequencies
for $\omega = q_0$, $\omega < F$, $\omega++$ **do**
  if $\delta(\omega, K_j)$'s frequency $< Threshold_f$ **then**
    Delete the $\delta(\omega, K_j)$
  end if
end for
for $\delta(q, K_j) < F$ **do**
  if $\delta(q, K_j)$ is special state **then**
    Delete the $\delta(q, K_j)$, Update the state transference $\delta(q+1, K_j)$
  else
    if $\delta(q, K_i) = \delta(q, K_j)$ **then**
      Delete the $\delta(q, K_j)$, Update the state transference $\delta(q+1, K_j)$
    end if
  end if
end for

Firstly the initial state q_0 is built. Then one of these messages are inputs to build one branch of this FSM. Every keywords is constructed as a transition

from the current state to a new state. New branch will be created when the first transition is undefined and a new path is created as well. When constructing the T, the probability of each transition is calculated.

FSM Generalization. The generalization concludes merging and reducing. We employ the Moore reduction procedure for deterministic finite automata minimization that generalize the FSM with the minimum number of states and transitions. The parameter $Threshold_f$ determines that keywords with low frequencies are useless. This condition will help us to filter the keywords again. Our framework will reduce or merge the node according to the principles.

Principle I: The node's frequency that lower than $Threshold_f$ will be deleted.

Principle II: The nodes which have same message type or the equivalent states will merge to one node. Create a new node to replace the other same nodes. If they are the equivalent ones, merge the circulate nodes into one. Then update the associative transition path and merge their probability at the same time.

After the minimization, the FSM becomes a concise one, but not generalization. To generalize the FSM, we should ensure that there is no special keyword like "time","date" in FSM. If it has special keywords, our framework will delete these nodes and update transfer paths. At last, we will update all the changed paths' probabilities and the generalized FSM will be treated as the $L - FSM$.

4.3 State Machine Building

In this component, the framework will build the $S-FSM$. State Machine Building Algorithm is described in Algorithm 3.

Construction of the Initial FSM. The protocol $S - FSM$ is constructed from the network sessions. The extracted sessions are based on the same timing semantics and the IP addresses etc. According to the languages we extracted, we can utilize these matched languages to stand for input messages.

The framework build a initial FSM S to cluster the keywords in the network sessions. We build the new state node when there is a distinct state appeared. Each transfer path is a whole complete session. Each node transition is a protocol language. The initial FSM will store all kinds of input sessions.

FSM Generalization. Our framework propose the principles to generalize the state FSM S into the concise one. After that we will the protocol $S - FSM$.

Principle I: The state nodes which have same language type will merge to one state node. Create a new state node to replace the other same state nodes, and update the associative transition path.

Principle II: The state nodes which have the equivalent states will be merged. Merge the circulate states nodes into one, update the associative transition paths.

(a) ARP (b) SMTP

Fig. 2. The Jaccard Index for ARP and SMTP

Algorithm 3. The State Machine Building algorithm

Input: The testing data set $T = \{T_1, T_2, ..., T_n\}$, the formats set $F = \{F_1, F_2, ..., F_n\}$
Output: The State Machine M

Create a dyadic array $S(m, 1)$
Let each T_i in testing data set $\rightarrow$ $S(i, 1)$, Label each $S(i, 1)$ with the format set $F = \{F_1, F_2, ..., F_n\}$
Delete the non-label parts in $S(i, 1) \rightarrow L(i, 1)$ and each $L(i, 1) = \{..., F_i, ..., F_j, ..\}$
Create a root *Root* for initial machine, Create a pointer p and $p \leftarrow Root$,the scan each $L(i, 1)$
for $L(i, 1) \neq NULL$ **do**
    Create a pointer L and $L \leftarrow$ the first element $L(0)$ in $L(i, 1)$
    for $L \neq NULL$ **do**
        if $p.Item_i = L$ **then**
            $p \leftarrow p.link_i$
        else
            Create a new state node N
            $N.Item \leftarrow L$, $p.link_i \leftarrow N.link$, $p \leftarrow N.link$
        end if
        $L \leftarrow L + 1$
    end for
    $i \leftarrow i + 1$
end for

5 Experiments and Evaluations

5.1 Data Set

In our experiments, the raw data comes from the DARPA (Defense Advanced Research Projects Agency) to insure the unbiased results. Our data set contains 4070 SMTP messages, 2386 ARP messages and 6000 other protocols messages. For binary protocol experiment, the raw data is the ARP messages. Each byte is considered as one token in this experiment. For text protocol experiment, the raw data is reprocessed to extract SMTP messages only. The TCP header and the IP header of the SMTP packets are already cut off.

5.2 Evaluation Metrics

In the evaluation experiments, we define the following three sets: True Positives: the set of protocol X messages(or sessions) where each one matches can be accepted by our framework. False Positives: the set of not protocol X messages

(or sessions) where each one can be accepted by our framework. False Negatives: the set of protocol X messages(or sessions) where each one can not be accepted by our framework. Next, the following two metrics are defined to quantitatively evaluate the effectiveness of our framework.

$$precision = \frac{|TruePositives|}{|TruePositives| + |FalsePositives|}$$

$$recall = \frac{|TruePositives|}{|TruePositives| + |FalseNegatives|}$$

5.3 Experimental Results

Binary Protocol. In binary protocol experiment, we only need to infer the protocol language for the methodology of binary protocol state which is same as text one. So we evaluate protocol state inferring in text protocol experiment. The ARP is adopted in our binary protocol experiment. Fig.2(a) presents the Jaccard index for ARP protocol. In this experiment we set the threshold $\lambda = 1500$. These frequent tokens are spliced into 5 keywords "$0xffffffffffffff$", "$0x08060001080006$", "$0x00$", "$0x01$" and "$0x000000000000$".

Table 1. The inferred SMTP protocol messages format

No.	Format	No.	Format	No.	Format
1	EHLO	2	250 pleased to	3	250 Sender
4	250 OK	5	250 Mail accepted	6	MAIL From:
7	RCPT To:	8	DATA	9	354 Enter mail .
10	220 Sendmail -0400	11	Received: -0400 Date: -0400	12	QUIT
13	221 connection	14	HELO	15	500

We construct the initial FSM T which is shown in Fig.3(a). The next step is the generalization. Like state S_1 to S_2, state S_0 to S_7, state S_9 to S_{10}, state S_0 to S_{13} and state S_0 to S_{16} share the same transition "$0x08060001080006$" offset 13, they are merged into state S_0 to S_1. The $L - FSM$ is shown in Fig.3(b).

(a) The Initial FSM T (b) Generalization

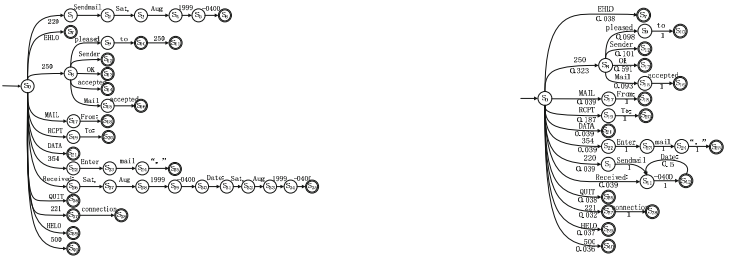
Fig. 3. The L-FSM for ARP

(a) The Precision Rate (b) The Recall Rate

Fig. 4. The Precision Rate and Recall Rate for Binary Experiment

The precision rate varying the number of input messages is shown in Fig.4(a). The value of precision rate is near 100% means few not ARP messages are accepted by our $L-FSM$. This rate measures the accuracy of the inference protocol language. The recall rate is shown in Fig.4(b). The value of recall rate is near 100% means that most of ARP messages are identified correctly. Therefore the $L-FSM$ is able to capture most of the ARP protocol language.

Text Protocol. 4070 SMTP messages are tokenized using " $< SP >$ " or " $< TAB >$ ". We set the keywords threshold $\lambda = 130$. The Jaccard index of SMTP is shown in Fig.2(b). Words like "Sender", "accepted" and date information are identified as keywords inaccurately because of their high frequencies. Fig.5(a) is the SMTP protocol language initial $FSM\ T$. During the generalization process, the keyword "−0400" appears from state S_5 to state S_6 , from state S_{29} to state S_{30}, from state S_{34} to S_{35} in Fig.5(b), has been merged into one transition from state S_{41} to state S_{42}. The keywords "Sat,", "Aug" and "1999" are deleted for they are inaccurate apparently. During minimization process, the keyword "250" appear from state S_{10} to state S_{11} is also filtered because of its low frequency. Fig.5(b) is the $L-FSM$ we inferred.



(a) The Initial FSM T (b) Generalization

Fig. 5. The $L-FSM$ for SMTP

(a) The Initial FSM S (b) Generalization

Fig. 6. The S-FSM for SMTP

(a) The Precision Rate (b) The Recall Rate

Fig. 7. The Precision Rate and Recall Rate for Text Experiment

After the sessions are extracted, each messages are replaced by their format number listed in Table 1 within one session. The initial FSM S is shown in Fig.6(a). Like transitions from q_5 to q_9 , from q_{20} to q_{25} and from q_{35} to q_{39} present exactly the same state transfer. States with the same transition are merged, such as the transitions from state q_2 to state q_{14} and from state q_{15} to state q_3 in Fig.6(a). The $S - FSM$ is presented in Fig.6(b).

We evaluate the precision and recall of our framework. Fig.7(b) is the recall rate. The value of recall rate is near 100% means that most of SMTP sessions are identified correctly. The precision rate is shown in Fig.7(a). The value of precision rate is near 97% means few not SMTP sessions are accepted by our $S - FSM$. This rate measures the accuracy of the inference protocol sate transition model.

6 Conclusion

In this paper, we propose a framework which can infer unknown protocols specifications and the state transferring machines based on the keywords identification and finite state machine construction. Our framework uses two different methods for binary protocols and text protocols without any priori information of the target protocols. We infer the protocol specification by constructing two FSMs. By keywords identification, we solve the time-consuming problem of the constructing FSM for some complex or long protocols. Our experimental results show that our framework is able to infer the language and the state machine of both binary and text protocols. The precision rate reaches 100% and the recall rate is 100% for ARP. The precision rate reaches 100% and the recall rate is 98% for SMTP. We are working towards extending our method to identify the fields mentioned above to improve the accuracy in the future.

Acknowledgment. We thank the anonymous reviewers for their helpful comments. This work is supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U1230106).

References

1. Caballero, J., Yin, H., et al.: Polyglot: Automatic extraction of protocol message format using dynamic binary analysis. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM (2007)
2. Antunes, J., et al.: Reverse engineering of protocols from network traces. In: 2011 18th Working Conference on Reverse Engineering, pp. 169–178. IEEE (2011)
3. Beddoe, M.: The protocol informatics project. Toorcon 4, 4 (2004)
4. Cui, W., Paxson, V., Weaver, N., et al.: Protocol-Independent Adaptive Replay of Application Dialog. In: NDSS (2006)
5. Leita, C., et al.: Scriptgen: an automated script generation tool for honeyd. In: 21st Annual Computer Security Applications Conference. IEEE (2005)
6. Weidong, C., Kannan, J., Wang, H.J.: Discoverer: Automatic protocol reverse engineering from network traces. In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium (2007)
7. Andrzejewski, D., et al.: A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol. 2, pp. 1171–1177. AAAI Press (2011)
8. Wondracek, G., Comparetti, P.M., Kruegel, C., et al.: Automatic Network Protocol Analysis. In: NDSS, vol. 8, pp. 1–14 (2008)
9. Lin, Z., Jiang, X., Xu, D., et al.: Automatic protocol format reverse engineering through conectect-aware monitored execution. In: 15th Symposium on Network and Distributed System Security, NDSS (2008)
10. Wang, Y., et al.: A semantics aware approach to automated reverse engineering unknown protocols. In: 2012 20th IEEE International Conference on Network Protocols (ICNP), pp. 1–10. IEEE (2012)
11. Du, W., et al.: Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 111–120. ACM (2010)
12. Trifilo, A., et al.: Traffic to protocol reverse engineering. Computational Intelligence for Security and Defense Applications (2009)
13. Xiao, M.M., et al.: Automatic Network Protocol Automaton Extraction. In: Network and System Security (2009)
14. De la Higuera, C.: Grammatical inference: learning automata and grammars. Cambridge University Press (2010)
15. Wang, Y., Zhang, Z., Yao, D(D.), Qu, B., Guo, L.: Inferring protocol state machine from network traces: A probabilistic approach. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 1–18. Springer, Heidelberg (2011)
16. Vidal, E., et al.: Probabilistic finite-state machines-part I. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(7), 1013–1025 (2005)
17. Hopcroft, J.E.: Introduction to Automata Theory, Languages, and Computation, 3rd edn. Pearson Education India (2008)

An Adaptive Collaborative Filtering Algorithm Based on Multiple Features

Yan-Qiu Zhang[1], Hai-Tao Zheng[2,*], and Lan-Shan Zhang[3]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhangyanqiu1001@163.com
[2] Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
zheng.haitao@sz.tsinghua.edu.cn
[3] Digital Media Art and Design Institute,
Beijing University of Posts and Telecommunications, Beijing, China
zls326@sina.com

Abstract. Due to the rapid development of E-commerce, personalized recommendations have been indispensable. The conventional user-based collaborative filtering (CF) cannot well satisfy users' requirements, besides the recommendation results are not accurate enough. To improve the conventional user-based CF, this paper proposes an adaptive CF method based on multiple features. We take four considerations into account: 1) redefining item-item/user-user similarity by utilizing item/user vector; 2) making predictions based on the relation between the predicted item and the rated similar items; 3) modifying the rating according to the interest in the type of item; 4) improving the diversity of recommendation. The proposed method is easy to implement, and experimental results based on two well-known datasets have demonstrated the superiority in accuracy and diversity.

Keywords: CF, item recommendation, user-user similarity, item-item similarity.

1 Introduction

Due to the rapid development of E-commerce, personalized recommendations have been indispensable. Recommendations include three classes: content-based recommendations, collaborative recommendations and hybrid approaches (Adomavicius et al. 2005), besides collaborative recommendation outperforms the others. CF means users collaborate to help one another perform filtering by recording their reactions to documents they read (Goldberg et al. 1999). CF-based recommendations are divided into two types: Memory-based (user-based) and Model-based (item-based) recommendations (Sarwar et al. 2001). User-based recommendations make recommendations based on a user's k-nearest neighbors. While item-based recommendations take root in item-item similarity. Item-item similarity is almost static and can be pre-computed offline, which can reduce the computational cost considerably.

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 455–465, 2013.

However, a single recommendation method has drawbacks inevitably. A single method only depends on single information, such as user-user/item-item similarity, which is poor for the prediction in recommendation accuracy. User-user similarity or item-item similarity is closely related to the users' attributes (e.g., age, gender, hobbies and the rated items) and the items' attributes (e.g., type and the rated number of users). In addition, the interests of a user may change with time, and the rating should vary with the user's interests. Furthermore, the relation between the predicted item and the similar rated items could make essential influence on the predicted item, which is often overlooked.

Based on user-based CF, this paper proposes an adaptive CF algorithm. The algorithm is executed by four steps: first, we build user vector and item vector; second, we modify the rating based on the principle of locality of time and user-item relation; third, we utilize user vector, item vector and the initial similarity to re-calculate user-user similarity and item-item similarity; finally, based on the above steps, and the relation between the predicted item and the similar rated items, we modify the prediction formula. And finally, we implement diverse recommendations by considering the most interesting item and the Top-N similar items.

The contributions of the paper include five aspects: 1) we redefine item-item similarity by utilizing item vector; 2) we utilize user vector including age, gender and interests to optimize user-user similarity; 3) we calculate the weight between the predicted item and the similar rated items to predict; 4) we modify the rating based on the rank of the item type in preference of users; 5) we implement diverse recommendation including the most popular item and the Top-N recommendations.

The remainder of the paper is organized as follows: section 2 introduces the related work concerning CF methods; the proposed method is explained in section 3; experimental results are discussed in Section 4; finally we conclude our work in section 5.

2 Related Work

Goldberg et al. 1999 design Tapestry to achieve the management of electronic mail documents. Tapestry supports CF framework and then many recommendation systems arise. At first, user-based CF recommendation systems achieve widespread success. However, two challenges restrain the usage: data sparsity and scalability. Many explorations are made to solve the problems. Ma et al. 2007 propose a novel method for missing data prediction. Ren et al. 2012 and Yin et al. 2013 present methods to solve data sparsity. Unlike user-based CF methods, we consider user-user similarity, item-item similarity and other factors to make recommendations.

Sarwar et al. 2001 propose item-based CF algorithms to eliminate the drawbacks of user-based CF algorithms. There are many explorations. Deshpande et al. 2004 combine item-based CF with the conditional probability-based similarity model to achieve better recommendations. Hidasi et al. 2013 propose context-awareness for item-item similarity in factorization framework. He et al. 2010 build a context-aware citation recommendation system to help users effectively cite related papers. Gedikli

et al. 2013 and Sen et al. 2009 combine tags to achieve recommendations. Park et al. 2006 improve the scalability and performance of a previous approach to implement robust cold-start recommendations using naïve filterbots. Unlike item-based CF methods, they mostly consider item-item similarity and use several factors (e.g., tag and context), we combine user-user similarity, item-item similarity, the attention of item, and the correlation between the predicted item and the similar rated items to recommend. These factors make the prediction more accurate and diverse.

There are other outstanding works. Xu et al. 2012 separate the user-item matrix into many sub-groups inspired by that one user may belong to multi-groups. Bellogin et al. 2013 present the method that selects neighbors according to the overlap of their preferences with those of the target user. Bartolini et al. 2011 propose CF Skyline to handle with the case that one item has several ratings. Park et al. 2007 propose a new ranking method which combines recommender systems with information search tools for better searching and browsing. CF methods are also used in network recommendation (Noel et al 2012, Niu et al. 2011 and Sandholm et al. 2011).

Based on the above user-based and item-based CF recommendation techniques, we propose our CF method to improve the recommendation accuracy and diversity.

3 The Adaptive Collaborative Filtering Algorithm

The algorithm is executed by six steps: 1) construct user and item vector; 2) modify the rating based on the interest in the item type; 3) compute user-user similarity; 4) compute item-item similarity; 5) modify the prediction formula; 6) implement diverse recommendation (Fig. 1).

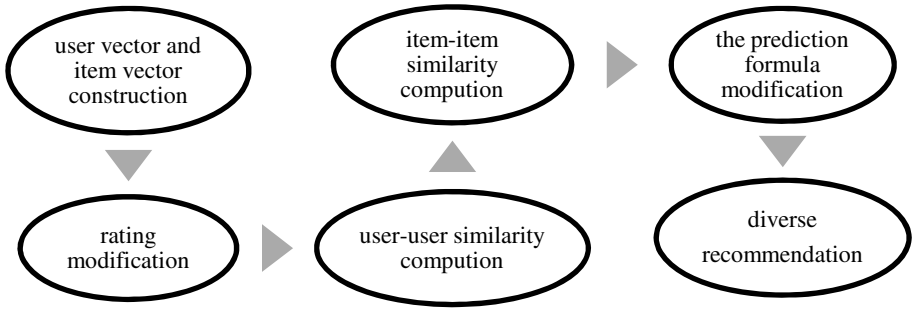


Fig. 1. The framework of the proposed method

3.1 User Vector and Item Vector

The conventional user-based/ item-based CF only considers ratings and overlooks the users'/items' attributes, therefore we construct the user /item vector integrated with the user-item matrix and user's/ item's attributes.

User Vector. The item type that a user likes affects the user-user similarity, and it's the same to the individuals that a user likes. The attributes of gender, age and hobbies can highly influence the user-user similarity. Besides, the rated number and the average rating also make effects. So we add the above features into user vector.

$$U = \{gender, age, type[], number, individual[]\} \tag{1}$$

$$gender = \begin{cases} 1, if\ gender = female; \\ 0, \qquad\qquad other \end{cases} \tag{2}$$

Variable U is the user vector; $type$ is the types that the user likes, by calculating the number of each type of items that the user has rated, then rank them to get the sequence of each type; $number$ is the rated number of items; $individual$ denotes the individuals that the user likes, by calculating the number of individuals related to the items that the user has rated, then rank them to get the sequence of each individual.

Item Vector. For item vector, except for the features (e.g., type and individuals), there are some other factors influencing item-item similarity (e.g., the rated number, the average rating, and the main age layer rating the item).

$$I = \{type[], number, individual[], age\} \tag{3}$$

Variable I denotes the item vector; type is the type that the item belongs to, and one item can belong to several types, besides the sequence is useless; $number$ denotes the rated number; $individual$ denotes the related individuals to the item (e.g., for the movie, it's the actor), and the sequence doesn't matter; age is the main age layer of individuals rating the item.

3.2 Rating Modification

The rating may vary with user's hobbies. User's hobbies may be static or vary with time. The recent behavior has vital effects for the later behavior, which means that if a user often watches one kind of movies recently, then in the next several days/months, he/she will possibly watch the same type of movies. The rating is based on the principle of locality to be modified. The modified rating is closely related to the rank of item type. This modification reasonably modifies the rating which can inflect the change of interests.

$$R_{u,i} = r_{u,i} \times (weight_i + Term) \tag{4}$$

$$weight_i = \gamma^{(1-\frac{Inumber}{Tnumber})} \tag{5}$$

$$Term = 1 - \gamma \tag{6}$$

Variable $R_{u,i}$ denotes the new rating; $r_{u,i}$ denotes the initial rating; $weight_i$ is the item weight in accord with the weight of item type; $Term$ is the term to balance the modified rating and the initial rating; $Tnumber$ denotes the total rated items; $Inumber$ is the item number of the type;γ is a constant value between$[0,1]$.

3.3 User-User Similarity Computation

There are three methods to compute user-user similarity: correlation-based similarity, cosine-based similarity, and adjusted cosine-based similarity. Sarwar et al. 2001 demonstrate the superiority of adjusted cosine similarity for MovieLens data. Our experiments are based on two well-known MovieLens datasets, so we use the adjusted cosine similarity to compute the user-user similarity:

$$UserSim(u,v) = \frac{\sum_{i\epsilon I_{uv}}(R_{u,i}-\overline{R_u})(R_{v,i}-\overline{R_v})}{\sqrt{\sum_{i\epsilon I_{uv}}(R_{u,i}-\overline{R_u})^2 * \sum_{i\epsilon I_{uv}}(R_{v,i}-\overline{R_v})^2}} \tag{7}$$

$UserSim(u,v)$ is the conventional user-user similarity; $R_{u,i}$ is the new rating; $\overline{R_u}$ is the average user's rating; I_{uv} is the items that have been rated by both users.

Due to the reasons that the attributes such as age, gender and user-item related data make influence on user-user similarity, we combine conventional user-user similarity with user vector similarity:

$$newUserSim(u,v) = \alpha UserSim(u,v) + (1-\alpha)USim(u,v) \tag{8}$$

Variable $newUserSim(u,v)$ denotes the new user-user similarity; $USim(u,v)$ is the user-user similarity computed by user vector according to cosine similarity; α is the weight between$[0,1]$.

3.4 Item-Item Similarity Computation

Sarwar et al. 2001 demonstrate the superiority of adjusted cosine similarity for MovieLens data, so we use the adjusted cosine-based similarity to compute item-item similarity:

$$ItemSim(i,j) = \frac{\sum_{u\epsilon I_{ij}}(R_{u,i}-\overline{R_u})(R_{u,j}-\overline{R_u})}{\sqrt{\sum_{u\epsilon I_{ij}}(R_{u,i}-\overline{R_u})^2 * \sum_{u\epsilon I_{ij}}(R_{u,j}-\overline{R_u})^2}} \tag{9}$$

Variable $ItemSim(i,j)$ is the conventional item-item similarity; I_{ij} denotes the users who have rated both items.

The conventional item-item similarity only relies on ratings, while some attributes like type and the number of individuals rating the item make effects. So we integrate conventional item-item similarity with item vector similarity:

$$newItemSim(i,j) = \beta ItemSim(i,j) + (1-\beta)ISim(i,j) \tag{10}$$

Variable $newItemSim(i,j)$ is the new item-item similarity; $ISim(i,j)$ is the item-item similarity computed by item vector utilizing cosine similarity; β is the weight between$[0,1]$.

3.5 Prediction Formula Modification

There are two methods to compute the predication score including weighted sum and regression; here we use the weighted sum to compute prediction score. The conventional user-based prediction formula is as follows:

$$p_{u,i} = \overline{R_u} + \frac{\sum_{v \epsilon N_u} newUserSim(u,v) * (R_{v,i} - \overline{R_v})}{\sum_{v \epsilon N_u} newUserSim(u,v)} \tag{11}$$

Variable $p_{u,i}$ denotes the prediction score that the user evaluates the item.

The prediction links to the attention of item, for instance, if the item is popular now, then it should be recommended. And the predicted item is related to the correlation between the item and the similar rated items.

$$newP_{u,i} = \lambda * \left(\overline{R_u} + \frac{\sum_{v \epsilon N_u} newUserSim(u,v) * (R_{v,i} - \overline{R_v}) * Sim(i,C_v)}{\sum_{v \epsilon N_u} newUserSim(u,v)}\right) * Sim(i,C_u) + (1-\lambda) * Attention(i) \tag{12}$$

Variable $newP_{u,i}$ denotes the new prediction score; $Attention(i)$ is the attention of item; $Sim(i,C_u)$ denotes the similarity between the item and the similar rated items; λ is the weight.

$$Sim(i,C_u) = \frac{\sum_{k=1}^{N} newItemSim(i,k)}{N} \tag{13}$$

Variable N denotes the Top-N similar items.

$$Attention(i) = \begin{cases} 5, if \exists newItemSim(i,j) = 1, j \in C_{popular} \\ 1, \qquad\qquad other \end{cases} \tag{14}$$

$$\lambda = \begin{cases} 0, if\ Attention(i) = 5 \\ 1, \qquad\qquad other \end{cases} \tag{15}$$

Variable $C_{popular}$ denotes the popular items at present, if the item is in the several first sets when ranking the average ratings or if the individual/type related to the item is the individual/type that the user likes, and then it's popular, besides it's easy to acquire the popular item sets. The recommendation can be diverse in several aspects: 1) gender; 2) age; 3) hobbies; 4) the Top-N recommendations according to Equation (12).

4 Evaluation

4.1 Experiment Data

We conduct experiments on two well-known datasets to evaluate our method: MovieLens 100K and MovieLens 1M. MovieLens 100K contains 943 users, 1682 movies and 100000 anonymous ratings, and yet each user has at least rated 20 movies. MovieLens 1M contains 6040 users and approximately 3952 items, and we extract a dataset including 2000 movies, 1414 users and 175813 anonymous ratings, and each user has at least rated 26 movies. The rating ranges from one to five. The higher the score is, the more the user likes the item. The datasets are basically adapted to our method, except that the datasets don't contain the individuals that the user likes.

We perform experiments on the effects of the number of user's/item's neighbors and γ to MAE (mean absolute error), and diversity results are also illustrated. In order to better demonstrate the superiority of our method, we compare our method with conventional user-based CF method and Weighted Slope One. During the process, we set all datasets as testing set and training set except that when comparing the differences between our method and Weighted Slope One. Variable α and β control the influence of user vector and item vector. User/Item vector is a small factor compared to the conventional user-user/item-item similarity, so we set α, β to 0.95.

4.2 Evaluation Metric

Accuracy Metric. There are several accuracy metrics including MAE and NMAE (normalized mean absolute error). MAE can well evaluate the prediction accuracy between the predicted rating and the real rating, thus we utilize MAE to evaluate the prediction accuracy. The smaller MAE is, the better the accuracy is.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \tag{16}$$

Variable n denotes the total number of recommendation items, p_i denotes the predicted rating, and q_i denotes the real rating.

Diversity Metric. By observing the Top-N list results, we can evaluate the diversity. When making predictions, the most popular items and the most interesting items should be recommended. To achieve the goal, we make specific modifications.

4.3 Experiment Results

The Number of Neighbors. The number of user's/item's neighbors affects MAE. In order to better examine the effects of the number of neighbors, we set the number of user's neighbors to 5, 10, 20, 30, and 40, and yet set the number of item's neighbors from 5 to 30.

When the number of user's neighbors is 5 and the number of item's neighbors is 20, the result is the best. For different values of item's neighbors, when the number is 20, the result is the best. For different values of user's neighbors, the smaller the number is, the better the result is (Fig. 2 left graph).

We conduct experiments on MovieLens 100K and MovieLens 1M, to better display the results when the number of item's neighbors is 20. The bigger the number of user's neighbors is, the bigger MAE is (Fig. 2 right graph). The results illustrate that the number of similar users is small, so we set the number of user's neighbors small to achieve better results in the other experiments.

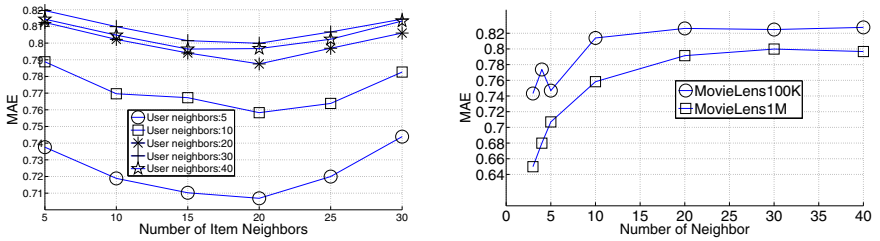


Fig. 2. MAE comparisons: the left is between different numbers of neighbors from MovieLens 100K; the right is for different numbers of user's neighbors when the number of item's neighbors is 20.

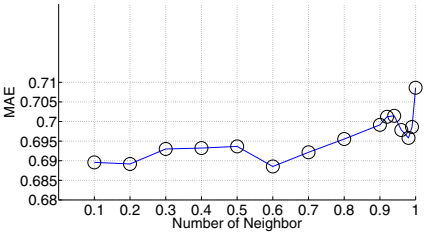


Fig. 3. MAE with different γ for MovieLens 100K

Variable γ Setting. Variable γ controls the speed of decay on rating. We perform experiments to evaluate the influence of γ (0 -- 1). During the experiments, we set the configuration to best: the number of user's neighbors is 5 and the number of item's neighbors is 20, and these settings make the results better than most of the other results in the experiments.

When $\boldsymbol{\gamma}$ is 0.6, the result is the best; when $\boldsymbol{\gamma}$ is too close to 0, the rating decays too fast; when $\boldsymbol{\gamma}$ is too close to 1, the rating decays too slowly (Fig. 3). Decaying too fast/slowly will make tiny improvements on MAE, so γ should be close to 0.5 to achieve better results.

Diversity Examination. For recommendations, diversity is equally important to accuracy. In most cases, we hope the recommendations are filled with diverse items.

When the score is 5.0, it means that the item is the most popular item at present or the user's favorite type, and then it should be recommended (Table 1).

Method Comparison. When comparing between the conventional user-based CF and our method, the number of user's neighbors is set from 5 to 40 or 10 to 40 and the number of item's neighbors is 20. In order to better demonstrate the effects of our method, we compare our method with Weighted Slope One for different numbers of test.

Table 1. The diverse results for some user

Rank	Score	Title
1	5.0	Independence Day(ID4) (1996)
2	5.0	Return of the Jedi(1983)
3	4.066219	Clockwork Orange, A(1971)
4	4.027505	Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963)
5	4.026662	Blade Runner (1982)
6	3.987419	Casablanca (1942)
7	3.972160	Shawshank Redemption, The (1994)
8	3.961542	Apollo 13 (1995)
9	3.946497	Die Hard (1988)
10	3.942902	Twelve Monkeys (1995)

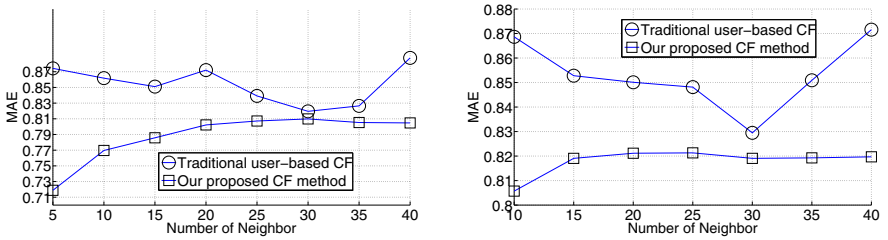


Fig. 4. MAE comparison between traditional user-based CF and our CF method for MovieLens 100K (the left graph) and MovieLens 1M (the right graph)

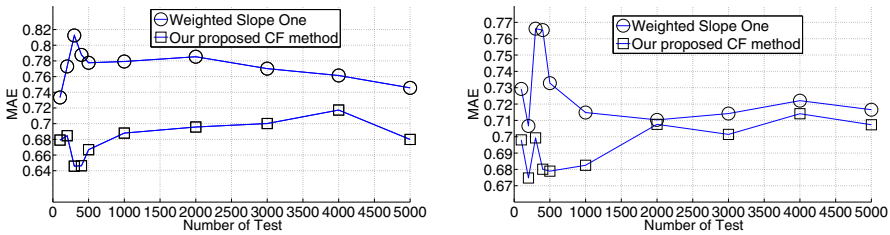


Fig. 5. MAE comparison between Weight Slope One and our method for MovieLens 100K (the left graph) and 1M (the right graph)

For different numbers of user's neighbors, our method outperforms the conventional user-based CF method in accuracy (Fig. 4). And for different numbers of test, our method is superior in accuracy compared to Weighted Slope One (Fig. 5). The results in Fig.5 are better than the results in Fig. 4 for MAE. In Fig. 4, we take all data as the dataset; however in Fig. 5, we only extract a subset of the dataset, which makes MAE smaller than the results in Fig. 4. Unlike the conventional user-based CF/Weighted Slope One which only considers user-user /item-item similarity, our method combines item-item similarity with user-user similarity, and modifies the ratings based on the change of user's interest and yet redefines user-user/item-item similarity by using user/item vector. Those improvements make our method better than the two methods in accuracy.

The overall results demonstrate that the proposed method outperforms in accuracy and diversity: 1) the preference of users to items may vary with time, thus we modify the ratings according to the degree of preference; 2) for user-user/item-item similarity, we consider ratings and attributes of users/items; 3) when making recommendations, the similarity between the item and the similar rated items is considered; 4) for diverse recommendation, we utilize the attention of an item.

5 Conclusion

In the paper, we aim to improve the recommendation accuracy and diversity. Based on the conventional user-based CF, we redefine user-user/item-item similarity utilizing user/item vector and calculate the predicted score. We consider the influence of substantial features, historical data of user and item, the correlation between the predicted item and the rated similar items, and the attention of the predicted item, and finally make diverse recommendations. Experiments are performed on two well-known datasets based on MAE metric and diversity results. The results demonstrate that our method is superior in accuracy and diversity.

Our method assumes that each user belongs to a single group, but in most cases, one user may belong to multiple groups. In the future, we will integrate multiple grouping with prediction.

Acknowledgements. This research is supported by the 863 project of China (2013AA013300), National Natural Science Foundation of China (Grant No. 61375054) and Research Fund for the Doctoral Program of Higher Education of China (Grant No.20100002120018).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)

3. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM (2001)
4. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46 (2007)
5. Ren, Y., Li, G., Zhang, J., Zhou, W.: The efficient imputation method for neighborhood-based collaborative filtering. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 684–693 (2012)
6. Yin, H., Sun, Y., Cui, B., Hu, Z., Chen, L.: LCARS: a location-content-aware recommender system. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 221–229 (2013)
7. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
8. Hidasi, B., Tikk, D.: Context-aware item-to-item recommendation within the factorization framework. In: Proceedings of the 3rd Workshop on Context-awareness in Retrieval and Recommendation, pp. 19–25 (2013)
9. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L.: Context-aware citation recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 421–430 (2010)
10. Gedikli, F., Jannach, D.: Improving recommendation accuracy based on item-specific tag preferences. ACM Transactions on Intelligent Systems and Technology (TIST) 4(1), 11 (2013)
11. Sen, S., Vig, J., Riedl, J.: Tagommenders: connecting users to items through tags. In: Proceedings of the 18th International Conference on World Wide Web, pp. 671–680 (2009)
12. Park, S.T., Pennock, D., Madani, O., Good, N., DeCoste, D.: Naïve filterbots for robust cold-start recommendations. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 699–705 (2006)
13. Xu, B., Bu, J., Chen, C., Cai, D.: An exploration of improving collaborative recommender systems via user-item subgroups. In: Proceedings of the 21st International Conference on World Wide Web, pp. 21–30 (2012)
14. Bartolini, I., Zhang, Z., Papadias, D.: Collaborative filtering with personalized skylines. IEEE Transactions on Knowledge and Data Engineering 23(2), 190–203 (2011)
15. Park, S.T., Pennock, D.M.: Applying collaborative filtering techniques to movie search for better ranking and browsing. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 550–559 (2007)
16. Noel, J., Sanner, S., Tran, K.N., Christen, P., Xie, L., Bonilla, E.V., Della Penna, N.: New objective functions for social collaborative filtering. In: Proceedings of the 21st International Conference on World Wide Web, pp. 859–868 (2012)
17. Niu, K., Chen, W., Niu, Z., Gu, P., Li, Y., Huang, Z.: A user evaluation framework for web-based learning systems. In: Proceedings of the Third International ACM Workshop on Multimedia Technologies for Distance Learning, pp. 25–30 (2011)
18. Sandholm, T., Ung, H.: Real-time, location-aware collaborative filtering of web content. In: Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation, pp. 14–18 (2011)
19. Liu, Y.J., Luo, X., Joneja, A., Ma, C.X., Fu, X.L., Song, D.: User-Adaptive Sketch-Based 3-D CAD Model Retrieval. IEEE Transactions on Automation Science and Engineering 10(3), 783–795 (2013)
20. Liu, Y.J., Fu, Q.F., Liu, Y., Fu, X.: A Distributed Computational Cognitive Model for Object Recognition. Science China (Series F: Information Sciences) (to appear, 2013)

Ensemble of Unsupervised and Supervised Models with Different Label Spaces

Yueyun Jin, Weilin Zeng, Hankz Hankui Zhuo, and Lei Li

Dept. of Computer Science, Sun Yat-sen University, Guangzhou, China
{jinyueyun1,zengwlin,zhuohank}@gmail.com, lnslilei@mail.sysu.edu.cn

Abstract. Ensemble approaches of multiple supervised and unsupervised models have been exhibited to be effective in predicting labels of new instances. Those approaches, however, require the label spaces of all supervised models to be identical to the target testing instances. In many real world applications, it is often difficult to collect such supervised models for the ensemble. In contrast, it is much easier to get large amounts of supervised models with different label spaces at a stroke. In this paper, we aim to build a novel ensemble approach that allows supervised models with different label spaces. Each supervised model is associated with an anomaly detection model. We view each supervised model as a *partial* voter and we manage to maximize the consensus between partial voting from supervised models and unsupervised models. In the experiments, we demonstrate the effectiveness of our approach in different data sets.

1 Introduction

Ensemble learning exploits multiple models to obtain better predictive performance than could be obtained from any of the constituent models [1]. Ensemble approaches such as bagging [2], boosting [3], and Bayesian model averaging [4], have exhibited their effectiveness in many applications. Based on previous ensemble learning technologies, approaches have been proposed to take advantage of unsupervised models, assuming testing instances are well clustered with unsupervised models in hand. Those approaches, such as Bipartite Graph-based Consensus Maximization (BGCM) algorithm [5], Locally Weighted Ensemble (LWE) [6], and C3E algorithm (Consensus between Classification and Clustering Ensembles [7]), have been shown to be effective in predicting labels of new instances.

Despite the success of previous ensemble approaches, they all assume label spaces of different supervised models are all identical to testing data. In real-world applications, it is often difficult to collect large amount of supervised models with identical label spaces to combine. *For example, the ubiquity of mobile phones and the increasing wealth of data generated from sensors and applications are giving rise to a new research domain across computing and social science. Issues in behavioral and social science from the Big Data perspective – by using*

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 466–477, 2013.

large-scale mobile data collected by users of mobile phones as input to characterize and understand real-life phenomena, including users' demographics – have drawn intense research interest [8]. To predict users' demographics, different supervised models will be trained with respect to different label spaces. For example, if the mobile data is collected from children's mobile phone, the supervised model in predicting children's age may have the label space from age 5 to 15; if the mobile data is collected from middle school students, the supervised model in predicting students' age may have the label space from age 10 to 20. These supervised models are probably trained by different research centers from different large mobile data collected from different groups of people. Our question is: given these supervised model in hand, can we predict the label of a new user whose age is probably from 5 to 20?

Combining supervised models with different label spaces is a difficult problem. For example, in Figure 1(a) model m_a can successfully separate instances with "Label 1" (indicating ages from 10 to 15) and "Label 2" (indicating ages from 5 to 10); and model m_b can successfully separate instances with "Label 1" and "Label 3" (indicating ages from 15 to 20) in Figure 1(b). When combining models m_a and m_b, instances with "Label 2" and "Label 3" will not be correctly separated, as shown in Figure 1(c), since m_a has a negative effect in predicting "Label 3" and m_b has a negative effect in predicting "Label 2", and it is difficult to decide which model should be used when a new instance coming.

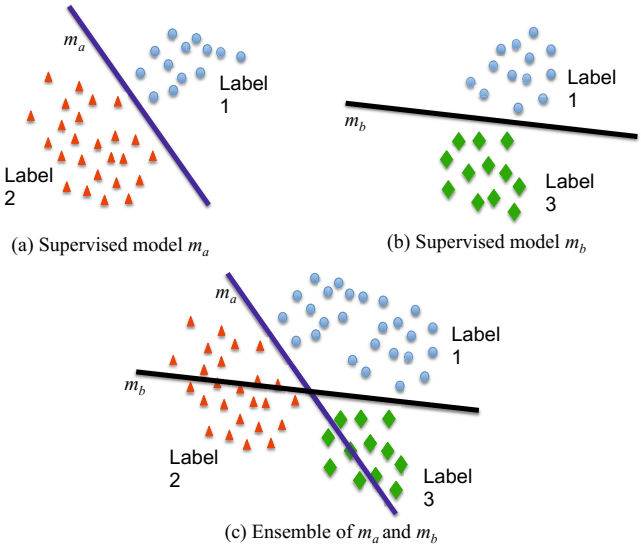


Fig. 1. An example of combining different supervised models, where Figure (c) is the result of combining Figures (a) and (b), "Label 1" suggests ages from 10 to 15, "Label 2" suggests ages from 5 to 10, "Label 3" suggests ages from 15 to 20

In this paper, we propose a novel approach, called ENUS, standing for **EN**semble of **U**nsupervised and **S**upervised models with different label spaces. We assume each supervised model is associated with an anomaly detection model [9] which views as anomalies the instances with labels that are not in the label space of the supervised model. As a result, we can exploit anomaly detection models to help decide which model can be used to predict labels of new incoming instances. We also assume there are unsupervised models (i.e., cluster models) available to help improve the prediction result. Although unsupervised models, such as clustering, do not directly generate label predictions, they provide useful constraints for the classification task. The rationale is that objects that are in the same cluster should be more likely to receive the same class label than the ones in different clusters. In summary, our ENUS approach functions by the following two phases. We first exploit anomaly detections to decide or choose which supervised models are to be used to classify new instances. After that we combine the chosen supervised models and unsupervised models to predict the labels of new instances.

We organize the rest of the paper as follows. We first review previous work related to our work in Section 2. After that we present our approach in detail in Section 3, and evaluate our approach in Section 4. Finally, we conclude the paper with future work in Section 5.

2 Related Work

Recent studies exhibit that unsupervised information can be leveraged to help improve the accuracy of supervised learning, such as semi-supervised [10] and transductive learning [11]. Semi-supervised learning (SSL) algorithms only take one supervised source (i.e., the labeled objects) and one unsupervised source (i.e., the similarity graph), and thus cannot be applied to combining multiple models. Some SSL methods [12] can incorporate results from a supervised model into a graph, but obviously they cannot handle multiple classifiers and unsupervised sources.

The ensemble of multiple supervised models has been proven to be more effective compared to the use of individual classifiers (or supervised models) [13]. Several research efforts have shown that cluster ensembles can improve the quality of results as compared to a single cluster (or unsupervised model) e.g., see [14] and references therein. Most of the motivations for combining ensembles of supervised and unsupervised models are similar to those that hold for the stand-alone use of either classifier or cluster ensembles. Additionally, unsupervised models can provide supplementary constraints for classifying new data and thereby improve the generalization capability of the resulting classifier. These successes motivate the design of effective ways of leveraging both classifier and cluster ensembles to solve challenging prediction problems.

However, approaches of combining classification and clustering models have been introduced only recently in the Bipartite Graph-based Consensus Maximization (BGCM) algorithm [5], the Locally Weighted Ensemble (LWE) algorithm [6] and, in the C3E algorithm (Consensus between Classification and

Clustering Ensembles [7]). Both BGCM and C3E have parameters that control the relative importance of classifiers and clusters. In traditional semi-supervised settings, such parameters can be optimized via cross-validation, which assume the distributions of training and testing data are identical. To remove this assumption, Acharya et al. [15] described a Bayesian framework, called BC3E, that takes as input class labels from existing classifiers (designed based on labeled data from the source domain), as well as cluster labels from a cluster ensemble operating solely on the target data to be classified, and yields a consensus labeling of the target data.

3 Our ENUS Algorithm

The framework of ENUS is shown in Algorithm 1. In step 2 of algorithm 1, we detect the anomaly instances for each classifier, then provide $\mathcal{X}^{dec}$ as input for the according classifier in step 3. In step 3, we produce the labels for $\mathcal{X}^{dec}$. After the Matrix B is obtained, we maximize the consensus in Algorithm 2.

Algorithm 1. An overview of our ensemble approach `ENUS`

input: A list of supervised models $\mathcal{M}^{sup}$, a list of unsupervised models $\mathcal{M}^{un}$, a list of anomaly detection models D, a list of training sets $\mathcal{X}^{train}$, a test set $\mathcal{X}^{test}$.
output: Labels matrix U of the test set $\mathcal{X}^{test}$.

1: **for** each supervised model $s \in \mathcal{M}^{sup}$ and the corresponding detection model d∈D **do**
2: use d to detect the test set $\mathcal{X}$ to get $\mathcal{X}^{dec}$ = d($\mathcal{X}$)
3: predict the label for test set $\mathcal{X}^{dec}$
4: **end for**
5: **for** each unsupervised models $s \in \mathcal{M}^{un}$ **do**
6: predict the group ID for test set $\mathcal{X}$
7: **end for**
8: associate the output obtained from supervised models and unsupervised model as matrix B
9: maximize the consensus between all supervised and unsupervised models by Algorithm 2;
10: **return** U;

Suppose we have a set of data points $X = \{x_1, x_2, \ldots, x_n\}$ from c classes. There are m models that provide information about the classification of X, where the first r of them are (supervised) classifiers, and the remaining are (unsupervised) clustering algorithms. But the label space of each classifier is different, which is smaller than or the same as the number of classes. Consider an example where $X = \{x_1, x_2, \ldots, x_7\}$, c=3 and m=4. The label space of M_3 is 1,2, while the M_4 is 1,3. The detection models give the anomaly instances label 'z'. The output of the four models are: $M_1 = \{1, 1, 1, 2, z, z, 2\}, M_2 = \{1, 1, z, z, z, 3, 1\}, M_3 = \{2, 2, 1, 3, 3, 1, 3\}, M_4 = \{1, 2, 3, 1, 2, 1, 1\}$, where M_1 and M_2 assign each object

Algorithm 2. Maximize the consensus

input: the number of supervised models $\mathcal{M}^{sup}$, the number of unsupervised models $\mathcal{M}^{un}$, group-object affinity matrix A, output matrix B of all models for a test set $\mathcal{X}$, parameter α and ξ.
output: labels matrix U of the test set $\mathcal{X}$.
algorithm:

1: initialize label matrix $U^{(0)}$ and $U^{(1)}$ randomly
2: compute $V^{(1)}$ according to Equ. 3
3: t = 1
4: while $||U^{(t)} - U^{(t-1)}|| > \xi$ do

$$\overrightarrow{q_{j\cdot}}^{(t+1)} = \frac{\sum_{i=1}^{n} a_{ij}\overrightarrow{u_{j\cdot}}^{(t)} + \alpha\overrightarrow{v_{j\cdot}}^{(t)}}{\sum_{i=1}^{n} a_{ij} + \alpha}$$

$$\overrightarrow{u_{i\cdot}}^{(t+1)} = \frac{\sum_{j=1}^{k} a_{ij}\overrightarrow{q_{j\cdot}}^{(t+1)}}{\sum_{i=1}^{k} a_{ij}}$$

$$t = t + 1$$

5: return $U^{(t)}$

a class label, whereas M_3 and M_4 simply partition the objects into three clusters and assign each object a cluster ID. Cluster(supervised) model partitions X into groups, and objects in the same group share the same cluster ID. We summarize the framework of this example in Figure 2(a),where "X" indicates a set of input test data, "Matrix B" indicates the output labels of the test data, "Matrix U" is the final label matrix, d1-d2 are two detection models, c1-c2 are two supervised models, c3-c4 are two unsupervised models.We summarize the data, and the corresponding output in Figure 2(b). In Figure 2(b), nodes at the left denote the groups output by the (m-r) models with unlabeled ones from the unsupervised models, nodes at the right denote the n objects, and a group and an object are connected if the object is assigned to the group by one of the models.

The objective is to predict the class label of $x_i \in X$, which agrees with the base classifiers' prediction, and meanwhile, satisfies the constraints enforced by the clustering models,as much as possible. To reach maximum consensus among all the models based on diverse label space,we define an optimization problem over the bipartite graph whose objective function penalizes deviations from the base classifiers' predictions, and discrepancies of predicted class labels among nearby nodes.

Suppose we have the output of r classification algorithms and s clustering algorithms on a data set X, and $r + s = m$. For the sake of simplicity, we assume that each point is assigned to only one class or cluster in each of the m algorithms, and the number of clusters in each clustering algorithm is c, the same as the number of classes. Label spaces of classifiers, however, are different from each other, which is often smaller than the output label space. Note that cluster ID z may not be related to class z. So each clustering algorithm partitions

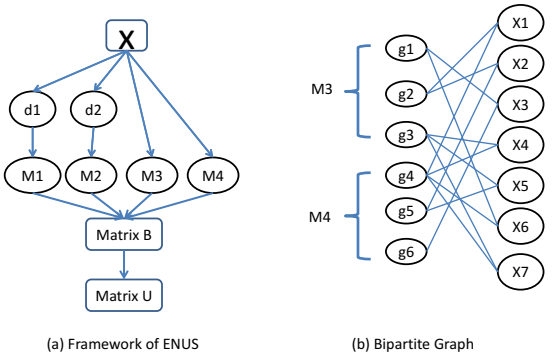


Fig. 2. An toy example of our ENUS algorithm, where Figure(a) is the process of Algorithm 1, Figure(b) is the bipartite graph for the groups and objects

X into c groups and there are a total of $k = sc$ groups, and all of them are from clustering algorithms. Before proceeding further, we introduce some notations that will be used in the following discussion: $B_{n\times m}$ matrix with b_{ij} representing the (ij)-th entry, and $\overrightarrow{b_{i.}}$ and $\overrightarrow{b_{.j}}$ denote vectors of row i and column j, respectively.

We represent the objects and groups in a bipartite graph as shown in Figure 2(b), where the object nodes $x_1, x_2, \ldots, x_n$ are on the right, the group nodes $g_1, \ldots, x_v$ are on the left. The affinity matrix $A_{n\times k}$ of this graph summarizes the output of s algorithms on X

$$a_{ij} = \begin{cases} 1, & \text{if } x_i \in g_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

We aim at estimating the conditional probability of each object node x_i belonging to c classes. As a nuisance parameter, the conditional probabilities at each group node g_j are also estimated. These conditional probabilities are denoted by $U_{n\times c}$ for object nodes and $Q_{k\times c}$ for group nodes, i.e.,

$$u_{iz} = \hat{P}(y = z|x_i),$$

and

$$q_{jz} = \hat{P}(y = z|g_j).$$

Since the label space of every classifier is not the same. The label space of the classifiers is either part of the whole label space or the same as the whole label space. $T(z)$ denotes the set of classifiers, the label space of which includes class z. Consider an example, there are three classifiers M_1, M_2, M_3 and one cluster M_4. The label space for M_1, M_2, M_3 are (1,2) ,(2,3),(1,3). So $T(1) = \{M_1, M_3\}$, and$|T(1)| = 2$. $V_{k\times c}$ matrix represents the ensemble conditional probabilities obtained from classifiers for group nodes:

$$s(j,z) = \sum_{x_i \in g_j} \sum_{t \in T(z)} \mathbf{1}(t(x_i) = z) \tag{2}$$

The function $\mathbf{1}(t(x_i) = z)$ is interpreted as below:

$$\mathbf{1}(t(x_i) = z) = \begin{cases} 1, & \text{if classifier t assigns } x_i \text{ to class z} \\ 0, & \text{otherwise} \end{cases}$$

After $s(j,z)$ is computed, v_{jz} is computed with a normalization term:

$$v_{jz} = \frac{s(j,z)}{\sum_{z=1}^{c} s(j,z)} \tag{3}$$

We formulate the consensus agreement as the following optimization problem on the graph:

$$\min_{U,Q} F(U,Q) = \min_{U,Q} \{ \sum_{i=1}^{n} \sum_{j=1}^{k} a_{ij} ||\overrightarrow{\boldsymbol{u}_{i.}} - \overrightarrow{\boldsymbol{q}_{j.}}||^2 + \alpha \sum_{j=1}^{k} ||\overrightarrow{\boldsymbol{v}_{j.}} - \overrightarrow{\boldsymbol{q}_{j.}}||^2 \} \tag{4}$$

s.t. $\overrightarrow{\boldsymbol{u}_{i.}} \geq \mathbf{0}, |\overrightarrow{\boldsymbol{u}_{i.}}| = 1, i = 1:n$; $\overrightarrow{\boldsymbol{q}_{j.}} \geq \mathbf{0}, |\overrightarrow{\boldsymbol{q}_{j.}}| = 1, j = 1:k$; $\overrightarrow{\boldsymbol{v}_{j.}} \geq \mathbf{0}, |\overrightarrow{\boldsymbol{v}_{j.}}| = 1, j = 1:k$

where $||.||$ and $|.|$ denote a vector's L2 and L1 norm respectively. The first term ensures that if an object x_i is assigned to group g_j by one of the clustering algorithm, their conditional probability estimates must be close. The second puts the constraint that a group g_j's consensus class label estimate should not deviate much from initial class label prediction. α is the shadow price payment for violating the constraints. Finally, $\overrightarrow{\boldsymbol{u}_{i.}}$ and $\overrightarrow{\boldsymbol{v}_{j.}}$ are probability vectors, and therefore each component must be greater than or equal to 0.

We propose to solve this problem using block coordinate descent methods as shown in Algorithm 2. At the t-th iteration, if we fix the value of U, the objective function is a summation of v quadratic components with respect to $\overrightarrow{\boldsymbol{q}_{j.}}$. Therefore it is strictly convex and λ.

4 Experiment

In this section, we demonstrate the effectiveness of the ENUS on three real-world applications. We construct seven different tasks and apply six classifiers and two clustering algorithms to every task. The classifiers can be randomly selected and so can the clustering algorithms. Note that, in the following experiments, the label spaces of all the classifiers is just part of the targets label space. That is to say, the training set's label space is part of the test set's label space. But the size of union set of all the classifiers' label space should be equal to the size of the target sets' label space. We assume that the accuracy of the anomaly detection model is nearly 100 percent in the following experiments. Below, we show you the details of the experiments.

4.1 Data sets and Experiment Setup

We evaluate our approach on three benchmark datasets, Pendigits, Letter, and Iris, by constructing seven tasks on these datasets.

Pendigits[1]: The size of Pendigits label space is ten. The objective is to classify instances under different label spaces. We randomly construct three tasks. For each task, the size of targets' label space is ten, but the size of classifiers' label space differs from each other among the tasks. And the label space of the classifier is part of the target's label space. We randomly separate the data set into training set and test set. The label space of the training set is modified by us manually. We learn Bayesian, J48 and IBK from the training sets, and apply these models as well as MakeDensityBaseCluster and SimpleKmeans on the targets.

Letter[2]: Aiming at classifying the image of letters, we extract three target sets to construct three tasks, each of which includes four kinds of letters. The size of targets'label space is four. The models we apply are shown in Tabel 3. The training sets' label spaces are different from the target sets, which are manually labeled according to Table 2. Both the training set and test set are extracted from the images of the letters.

Iris[3]: Our goal is to predict the label for the target sets. The data set is randomly separated into target set and training set. And the label space of the training set is just part of the target's label space. One task is constructed on this application. For the task, we set the whole six classifiers to be IBK and the clustering algorithm to be MakeDensityBaseCluster and SimpleKmeans.

The details of each task are summarized in Table 1, Table 2 and Table 3. Table 1 is data set description. The ID represents the number of the task for every application. There are seven tasks in all, three of which belong to the first application, while another three belong to the second application and the remaining one belongs to the third one. Table 2 is the label space description of classification models. There are 6 classifiers in all for each task. We denote the label space of the target set as A and denote the label space of classification model as B, which is part of A. It means that the classification model can only predict the instance where its label belongs to B. If the label of the instance belongs to A-B, the anomaly detection model can predict an anomaly label to it, which is labeled as z. For example,to the data set Pendigits, the the label space for the first task of six classifiers is {M1:01234, M2:56789, M3:56, M4:789, M5:012, M6:3456}. So M1 has the ability to predict the instance where the label belongs to "01234" and the anomaly label z. For the three tasks of Pendigits, the target set is the same. What is different is the label space of the six classifiers for each task.For example, the label space for the first task of six classifiers is {M1:01234, M2:56789, M3:56, M4:789, M5:012, M6:3456} while the label space of the second task is {M1:23456, M2:01789, M3:0123, M4:4567, M5:079, M6:068}. The label space is randomly selected but we should make sure that the union set of the six classifiers' label space covers the whole label of the target set.

[1] `http://archive.ics.uci.edu/ml/machine-learning-databases/pendigits/`
[2] `http://archive.ics.uci.edu/ml/datasets/Letter+Recognition`
[3] `http://archive.ics.uci.edu/ml/datasets/Iris`

For the three tasks of the Letter data set, the training sets and target sets are different. The same is true for the selection of the label space of the last four tasks. Table 3 is the classification and cluster models of the tasks. M1-M6 are classification models and M7-M8 are cluster models. In principle, we can choose the classification models and cluster models randomly. The tasks of Pendigits and Letter have different classification models while the classification models of the last task are the same.

Table 1. Data Sets Description

Data set	ID	CATEGORY Labels	Training-set	Target
Pendigits	1	{ 0,1,2,3,4,5,6,7,8,9}	6,000	2,000
	2	{ 0,1,2,3,4,5,6,7,8,9}	6,000	2,000
	3	{ 0,1,2,3,4,5,6,7,8,9}	6,000	2,000
Letter	1	{ A,C,F,W}	1,548	1,504
	2	{ A,L,V,Z}	1,847	1,200
	3	{ A,J,RU}	1,906	1,200
Iris	1	{Iris-setosa,Iris-versicolor, Iris-virginica}	60	90

Table 2. Label Space of Classifiers

Data set	ID	M1	M2	M3	M4	M5	M6
Pendigits	1	{0,1,2,3,4}	{5,6,7,8,9}	{5,6}	{7,8,9}	{0,1,2}	{3,4,5,6}
	2	{2,3,4,5,6}	{0,1,7,8,9}	{0,1,2,3}	{4,5,6,7}	{0,7,9}	{0,6,8}
	3	{3,4,5,6,7}	{0,1,2,8,9}	{0,2,4,6}	{1,3,5,7}	{2,8}	{1,6,7}
Letter	1	{A,C}	{F,W}	{A,C,F}	{C}	{A,F,W}	{A,W}
	2	{A,V,Z}	{A,L,V}	{V,Z}	{A,Z}	{A,L}	{L}
	3	{A,R,U}	{A,J,R}	{R,U}	{A,U}	{A,J}	{J}
Iris	1	{(I)}	{(I), (II)}	{(I), (III)}	{(II)}	{(II)}	{(II), (III)}

Note:(I) Iris-setosa, (II) Iris-versicolor, (III) Iris-virginica

In every task, we apply eight models from M1 to M8. The first six models are classifiers and the remaining are clustering models. The proposed ENUS combines the output of the six anomaly detection models,six classifiers and two clustering models to produce a reasonably better ensemble result. On each task, we repeat the experiments 3 times, each of which has a randomly chosen target set, and report the average accuracy.

4.2 Experimental Results

In this part, we report the experimental results regarding the effectiveness of the ENUS algorithm. The results clearly demonstrate that on the different classifiers label space problem , the proposed ENUS method can achieve a better

Table 3. Classifier and Cluster Models of Each task

Data set	ID	M1	M2	M3	M4	M5	M6	M7	M8
Pendigits	1	Bayesian	Bayesian	J48	J48	IBK	IBK	(I)	(II)
	2	IBK	Kstar	Bagging	RandomForest	J48	FT	(II)	(I)
	3	IBK	Kstar	Bagging	RandomForest	J48	FT	(II)	(I)
Letter	1	IBK	Kstar	Bagging	AdaBoosting	RF	J48	(II)	(I)
	2	Bayesian	Bayesian	J48	J48	IBK	IBK	(I)	(II)
	3	Bayesian	Bayesian	J48	J48	IBK	IBK	(I)	(II)
Iris	1	IBK	IBK	IBK	IBK	IBK	IBK	Kmeans	Kmeans

Note:(I)MakeDensityBaseCluster, (II) SimpleKmeans

Table 4. Classification and Cluster Accuracy Comparison on Seven Tasks

Model	Pendigit-1	Pendigit-2	Pendigit-3	Letter-1	Letter-2	Letter-3	Iris-1
M1	0.4237	0.4940	0.4917	0.5026	0.7025	0.7481	0.3333
M2	0.3998	0.4713	0.4750	0.4946	0.6783	0.7469	0.6519
M3	0.1755	0.3815	0.3808	0.73997	0.4922	0.4658	0.6333
M4	0.2908	0.3657	0.3632	0.2395	0.4881	0.4500	0.3111
M5	0.3015	0.2678	0.1852	0.7432	0.4997	0.48007	0.3111
M6	0.3927	0.2743	0.2767	0.4894	0.2497	0.2392	0.6222
M7	0.7483	0.7585	0.7585	0.8265	0.7911	0.7486	0.8667
M8	0.7585	0.7623	0.7630	0.8358	0.8019	0.7497	0.8963
4-ENUS	0.7385	0.7618	0.7587	0.8291	0.8014	0.7533	0.8704
6-ENUS	0.6902	0.7540	0.7508	0.8291	0.8014	0.7533	0.8889
8-ENUS	0.7515	0.7645	0.7485	0.8291	0.8014	0.7533	0.8704

Table 5. Sensitivity of Parameter α

Task	α=2	α=3	α=4	α=5	α=10	α=15	α=20
Pendigits-1	0.7515	0.7515	0.7515	0.7515	0.7520	0.7520	0.7660
Letter-1	0.8291	0.8291	0.8291	0.8291	0.8291	0.8291	0.8291
Iris-1	0.8704	0.8704	0.8704	0.8704	0.8704	0.8704	0.8704

accuracy.To compare the performance of the proposed ENUS framework, we evaluate the proposed framework in three different aspects.

Accuracy. In Table 4, the first six classifiers usually have low accuracy while the proposed ENUS algorithm always outperforms the six classification models and reaches a better performance. As to the relation of the cluster accuracy and the proposed ENUS algorithm, the higher the accuracy of the cluster model is, the better the result of ENUS method is. The result demonstrates the power of the ENUS method in accuracy improvement. In Table 4, the first six single classifier usually has low accuracy while the proposed ENUS algorithm always outperforms the six classification models and reaches a better performance. As to

the relation of the cluster accuracy and the proposed ENUS algorithm, the higher the accuracy of the cluster model, the better the result of ENUS method. The result demonstrates the power of the ENUS method in accuracy improvement.

Number of the Models. We vary the number of classifiers and cluster algorithms incorporated into the ensemble framework in our experiments. The ENUS method on two classifiers and cluster algorithms is denoted as 4-ENUS, where we average the performance of the combined model obtained by randomly selecting two classifiers and two cluster algorithm. One point needed to confirm is that the union set of the classifiers label spaces are the same as the target set. Similarly, ENUS method of four classifiers and two cluster algorithms is denoted as 6-ENUS. And ENUS method on six classifiers and two clusters is denoted as 8-ENUS. From Table 4, we can see that in most cases, 6-ENUS and 8-ENUS outperform 4-ENUS.When the base classification models are independent and each of them obtains reasonable accuracy, combing more models would benefit more because the chances of reducing independent errors increase. However, when the base classification model cannot obtain reasonable accuracy, combining the model would decrease the accuracy of ENUS accuracy. What is more, when the new classification model cannot provide additional information to the current set of models, incorporating it may not improve the performance anymore. Similarly, clusters contribute to the ensemble method in the same way. If the cluster can obtained reasonable accuracy, it can increase the ensemble accuracy by reducing independent errors.

Sensitivity. There is one important parameter in the proposed algorithm α. As shown in Table 5, the proposed method ENUS is not sensitive to the parameters α. α is the shadow price paid for deviating from the estimated labels of groups so α should be greater than 0. α represents the confidence of the of our belief in the labels of the groups compared with 1. The label of groups are obtained from classification models and may not be correct. So need to set a shadow price paid to achieve better performance. In order to make the experiments results more clear, we just report the average performance of the first task of three data set. In our experiments, we let $\alpha = 2$ to get the experiments results shown in Table 5. We could also note that in spite of the changes caused by parameter variation, the proposed ENUS improves over almost all of the base classification model greatly.

5 Conclusion

In this paper, we propose to combine the complementary predictive powers of multiple supervised and unsupervised models to derive a consolidated label assignment for a set of instances. Our approach allows the label spaces of supervised models to be different, which makes the ensemble problem more difficult. In this paper, we assume each supervised model is associated with an anomaly detection model. In this paper, we assume each supervised model is associated

with an anomaly detection model of high-quality. In the future we would like to test the influence of anomaly detection model with various accuracies. We also would like to explore the possibility of removing anomaly detection models.

Acknowledgement. Yueyun Jin and Hankz Hankui Zhuo thank the National Natural Science Foundation of China (No. 61309011) for supporting this research.

References

1. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. JAIR 11, 169–198 (1999)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Freund, Y.: Boosting a weak learning algorithm by majority. Inf. Comput. 121(2), 256–285 (1995)
4. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: a tutorial. Statistical Science 14(4), 382–401 (1999)
5. Gao, J., Liang, F., Fan, W., Sun, Y., Han, J.: Graph-based consensus maximization among multiple supervised and unsupervised models. In: Proceedings of NIPS, pp. 585–593 (2009)
6. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD, pp. 283–291 (2008)
7. Acharya, A., Hruschka, E.R., Ghosh, J., Acharyya, S.: C^3E: A framework for combining ensembles of classifiers and clusterers. In: Sansone, C., Kittler, J., Roli, F. (eds.) MCS 2011. LNCS, vol. 6713, pp. 269–278. Springer, Heidelberg (2011)
8. Laurila, J.K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.M.T., Dousse, O., Eberle, J., Miettinen, M.: The mobile data challenge: Big data for mobile computing research. In: MCSMobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing (2012)
9. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. 41(3) (2009)
10. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, Univ. of Wisconsin-Madison (2005)
11. Joachims, T.: Transductive learning via spectral graph partitioning. In: ICML, pp. 290–297 (2003)
12. Goldberg, A.B., Zhu, X.: Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In: First Workshop Graph Based Methods for Natural Language Processing (2006)
13. Oza, N.C., Tumer, K.: Classifier ensembles: Select real-world applications. Information Fusion 9(1), 4–20 (2008)
14. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. Statistical Analysis and Data Mining 4(1), 54–70 (2011)
15. Acharya, A., Hruschka, E.R., Ghosh, J., Sarwar, B.: Probabilistic combination of classifier and cluster ensembles for non-transductive learning. In: SDM (2013)

Cost-Sensitive Extreme Learning Machine

Enhui Zheng[1,*], Cong Zhang[1], Xueyi Liu [2], Huijuan Lu[3], and Jian Sun[1]

[1] College of Mechanical and Electrical Engineering,
China Jiliang University, Hangzhou 310018, China
[2] Department of Mathematics, China Jiliang University, Hangzhou 310018, China
[3] College of Information Engineering, China Jiliang University, Hangzhou 310018, China
ehzheng@cjlu.edu.cn

Abstract. ELM is an effective machine learning technique, which works for the "generalized" single-hidden-layer feed-forward networks. However, like original SVM, ELM and majority of its variants have been extensively used in classification applications. Compared to SVM , ELM achieve optimal solutions and require lower computational complexity. More and more researchers have been attracted by ELM due to its fast learning speed and excellent generalization performance. Traditional ELM presumes higher accuracy based on the assumption that all classes have same cost, and the sample size of each class is approximate equal. However, the assumption is not valid in some real cases such as medical diagnosis, fault diagnosis, fraud detection and intrusion detection.

To deal with classification applications where the cost of errors is class-dependent. we propose a cost-sensitive ELM Experimental results using classification data show that CS-ELM is effective.

Keywords: Extreme learning machine, Cost-sensitive, Classification.

1 Introduction

Extreme Learning Machine (ELM), recently proposed by Huang et al. [1-5], is a kind of effective and efficient learning algorithm for single-hidden-layer feed-forward networks (SLFNs), which has attracted the attentions from more and more researchers[6-15]. Different from the error back propagation (EBP) algorithm that adopts time-consuming iterative method, ELM first randomly generates and fixes the parameters of its hidden layer (input weights and bias), then the output weights are analytically determined by the least-squares method. Due to its similar or better generalization performance and much faster learning speed than Support Vector Machines (SVMs) and EBP, many works [8] have been devoted to ELM, and tremendous improved versions of ELM are presented to further enhance its performance [7].

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 478–488, 2013.

However, ELM and majority of its improved versions first implicitly assume that the errors of all training samples have the same cost, and then attempt to yield the maximal classification accuracy, which could be invalid when the misclassification cost are unequal. Actually, the above assumption is far from the case in so many real-world data mining fields as medical diagnosis, intrusion detection, fault diagnosis, and so on, where the misclassification of each positive sample is usually more expensive than that of each negative one [16]. Weighted regularized ELM was presented by [6], [9], and [10]. However, none of them was targeting the cost-sensitive learning problem, although [6] stated that classification applications where the cost of errors is class-dependent. Could be generalized to cost-sensitive learning. Further, previous work is based on multiply the error $e_{j,m}$ of the row of the matrices , while in this paper, multiply the error $e_{j,m}$ of the columns of the matrices . In fact, cost-sensitive learning has already attracted much attention from the machine learning and data mining communities[16]. In this paper, we focus on the classification problem, and the class-dependent cost is considered. To minimize total cost rather than total error rate, many cost-sensitive learning methods have been developed [17]. In those methods, generally, the performance of the resultant classifier is evaluated in test phase according to the class-dependent cost that is extracted and is fixed by the domain experts.

In this paper, inspired by the cost-sensitive learning, we propose the cost-sensitive version of ELM called cost-sensitive ELM (C-ELM) to tackle binary/multiclass classification tasks with class-dependent cost. C-ELM maintains some important features from ELM, such as ELM provides a unified learning platform with a widespread type of feature mappings and can be applied in regression and multiclass classification applications directly; ELM can approximate any target continuous function and classify any disjoint regions; ELM tends to have better scalability and achieve similar (for regression and binary class cases) or much better (for multiclass cases) generalization performance at much faster learning speed (up to thousands times) than SVM.

This paper is organized as follows. Section II briefly reviews the ELM. In Section III, the cost-sensitive versions of ELM are proposed from both algorithmic view and optimization view. In section IV, we discuss some issues related to C-ELM. The experiments are conducted in section V, and section VI concludes this paper.

2 ELM

2.1 Least-Square Based ELM

ELM is a kind of learning algorithm for SLFNs [3]. The conventional EBP algorithm can also be employed to train the SLFNs. the key advantage of ELM is its much faster learning speed than SVMs or EBP with similar or better generalization performance. A review of ELM can be found in [5].

In this section, we investigate the classification problems based on ELM. A set of training samples $X = \{(x_j, y_j)\}_{j=1}^{N}$, where $x_j = \left\{x_{j1}, x_{j2}, ..., x_{jd}\right\} \in R^d$, where

$y_j \in [1, M]$ each class label is expanded into a label vector of length m, For an *M*-labels classification application, the output label y_i of a sample x_j is usually encoded to an *M*-dimensional vector $(y_{j1}, y_{j2}, ...y_{jM})^T$ with

$$y_{jm} \in (1,0)(m = 1,2,...M). \tag{1}$$

the output of the SLFN with *L* hidden nodes and an activation function $g(x)$ (infinitely differentiable demanded) can be calculated by

$$\left[o_{j,1}, o_{j,2}, o_{j,m}, ..., o_{j,M}\right] = \sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x_j),\ j = 1, \cdots, N. \tag{2}$$

If the SLFN in (2) can approximate those N samples with zero error, there exits β_i, a_i and b_i such that

$$\left[y_{j,1}, y_{j,2}, y_{j,m}, ..., y_{j,M}\right] = \sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x_j),\ j = 1, \cdots, N \tag{3}$$

where $a_i \in R^d$ (the input weights connecting the input neurons and the hidden neurons) and $b_i \in R$ (hidden bias), $i = 1, \cdots, L$, are the hidden nodes' parameters that are randomly generated, and β_i, $i = 1, \cdots, L$, denote the output weights connecting the *i-th* hidden node to the output layer. Those *N* equations in (3) can be compactly written as

$$Y = H\beta, \tag{4a}$$

where

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times \mathrm{M}},\quad H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \vdots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L}, \text{ and}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times \mathrm{M}}. \tag{4b}$$

In the equation (4), H is the hidden layer output matrix, and $h(x)$ denotes the hidden layer feature mapping of SLFNs. If $L = N$, there exists a SLFN that can fit the training samples with zero error [17]. Therefore, given H and Y.

2.2 Least-Square Based ELM

An alternative approach for multiclass applications is to let ELM have multioutput nodes instead of a single-output node. m-class of classifiers have m output nodes.

If the original class label is p, the output label t_i of a sample x_j is usually encoded to an M-dimensional vector $(t_{j1}, t_{j2}, ...t_{jM})^T$ with $t_{jm} \in (1,0)(m = 1,2,...M)$ .The classification problem for ELM with multioutput nodes can be formulated as

$$\text{Minimize}: \ L = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{j=1}^{N}\|e_j\|^2 \tag{5}$$

$$\text{Subject to}: \ h(x_j)\beta = t_j^T - e_j^T, j = 1,...,N, \ e_j \geq 0.$$

According to KKT theorem, the equivalent dual optimization problem with respect to (7) is :

$$L = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{j=1}^{N}\|e_j\|^2 - \sum_{j=1}^{N}\alpha_j(h(x_j)\beta - t_j + e_j) \tag{6}$$

where the Lagrange multiplier α_j, is the constant factor of sample Xi in the linear combination to form the final decision function.

The decision function is

$$f(x) = h(x)\hat{\beta} = h(x)(H^{\dagger}Y). \tag{7}$$

The ELM algorithm can be summarized as Table 1.

Table 1. The Step of ELM Algorithm [5]

Algorithm: ELM
Input: A training set $\{(x_j, y_j)\}$, $j = 1,2,...,N$, where $x_j \in R^d$ and $y_j \in R^2$, hidden node output function $g(x)$, and the number of the hidden nodes L
Output: Decision function $f(x)$
Step: (1) Randomly generate hidden node parameters, $\{a_i, b_i\}$, $i = 1,2,...,L$
(2) Calculate the hidden layer output matrix H
(3) Calculate the output weight $\beta = H^{\dagger}Y$
(4) Obtain decision function $f(x) = h(x)\hat{\beta} = h(x)(H^{\dagger}Y)$

3 Cost-Sensitive Extreme Learning Machine

For an M-labels classification application, the goal is also to minimize the cumulative error $e_{j,i}$ .In C-ELM algorithm we multiply the column of error matrix e_j of the sample x_j by the $w_{j,m}(m = 1,2,...M)$ like weighted ELM:

$$\text{Minimize}: \ L_{p_{ELM}} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{j=1}^{N}\left\|e_j w_{jm}\right\|^2 \tag{8}$$

$$\text{Subject to}: \ h(x_j)\beta = y_j^T - e_j^T, j = 1,...,N$$

According to KKT theorem, the equivalent dual optimization problem with respect to (14) is :

$$L = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{j=1}^{N}\left\|e_j w_j\right\|^2 - \sum_{j=1}^{N}\alpha_j(h(x_j)\beta - t_j + e_j) \tag{9}$$

where the Lagrange multiplier α_j, is the constant factor of sample Xi in the linear combination to form the final decision function. Further, by making the partial derivatives with respect to variables $(\beta, e, , \alpha_j)$ all equal to zero, the KKT optimality conditions are obtained

$$\frac{\partial L}{\partial \beta} = 0 \to \beta = \sum_{j=1}^{N}\alpha_j h(x_j)^T = H^T\alpha$$

$$\frac{\partial L}{\partial e} = 0 \to \alpha_j = Ce_j w_j, \quad j = 1,...,N \tag{10}$$

$$\frac{\partial L}{\partial \alpha_j} = 0 \to h(x_j)\beta - t_j + e_j = 0, \quad j = 1,...,N$$

Therefore, the corresponding decision function is depicted as

$$f_C(x) = h(x)\hat{\beta}_C = h(x)(H^{\dagger}(YW_C)) = h(x)(H^{\dagger}Y)W_C \tag{11}$$

The CC-ELM-AC algorithm can be summarized as Table 2.

Table 2. The step of C-ELM Algorithm

Algorithm: C-ELM
Input: as done in section ELM.
Output: Decision function $f_C(x)$
Step: (1) Randomly generate hidden node parameters, $\{a_i, b_i\}\ i = 1,2,...,L$
(2) Calculate the hidden layer output matrix H
(3) Calculate $Y_F = YW$.
(4) Calculate the output weight $\hat{\beta}_C = H^{\dagger}Y_F$
(5) Obtain decision function $h(x_i)\beta = y$.

4 Performance Evaluation

Assume a set of training samples $X = \{(x_j, y_j, C_j)\}_{j=1}^{N}$, where Actual Cost $C_{jm} = \{C_{j1}, C_{j2}, C_{jm}, ..., C_{jM}\}$, $y_j \in [1, M]$ each class label is expanded into a label vector of length m , For an *M*-labels classification application, $C_m, C_m \in [C_1, ...C_M]$ are the misclassification cost of sample x_j . Given a parameter C (Actual Cost) and the training set X_{tr} , we can learn the decision model $f(x, C, X_{tr}) \in H$, The total cost is calculated by

$$L(C_m, P_C(m)) = \frac{1}{M} \sum_{m=1}^{M} C_m P_C(m) \tag{12}$$

where $P_C(m), (m = 1, 2, ...M)$ is called the classification accuracy of every classes with Actual Cost's decision model.

The goal is to find a classifier $f(x)$ minimizing the total cost. .Different CP lead to different decision models, and different decision models result in different total cost in test datasets. Therefore, the optimal value of the CP (or the optimal decision model) is obtained by minimizing the total cost in test datasets.

5 Discussions

Firstly, a comparison is conducted between ELM and the proposed method and a comparison is conducted between other weight ELM and the proposed method. Secondly, the consistence of C-ELM and cost-sensitive version of ELM by threshold moving can be proved. Finally, Issues about how to interpret the effect of the cost parameter and determine the value are discussed as well. The effect of adding the cost matrix to classification performance improvement is illustrated by classification problem.

5.1 The Difference between C-ELM and ELM

In this subsection, we use the C-ELM algorithms, to classify the two-class samples that are random generated and are depicted in Fig 1. In ELM, we set (C^+, C^-) as $(1,1)$, while in the C-ELM algorithms, we set (C^+, C^-) as $(2,1)$ and $(5,1)$, respectively. The trained decision hyper-planes of the ELM and the three algorithms are compared, and the results is illustrated in Figure 1 respectively. It can be seen that given the $C^- = 1$, with the increase of C^+ from 1 to 2, and to 5, generally, the decision hyper-planes of the proposed algorithms respectively move away from the positive samples that is more expensive than the negative ones.

C-ELM algorithms with cost matrix are presented, in which the cost matrix not only is directly integrated into the training process or decision model of the ELM, but also is used to evaluate the performance of the resultant classifier in test phase. As discussed in [30], it cannot be guaranteed that the minimal total cost is obtained by directly embedding the cost matrix into the training process or decision model of the ELM.

In this section, we incorporate the parameterized cost matrix , termed as cost parameter (CP), rather than the AC itself into the ELM algorithm or its decision model, but evaluate the performance of the resultant classifier according to the cost matrix. Different CP lead to different decision models, and different decision models result in different total cost in test datasets. Therefore, the optimal value of the CP (or the optimal decision model) is obtained by minimizing the total cost in test datasets. This paper uses the grid-search method to search Cost Parameter that minimizes total cost.

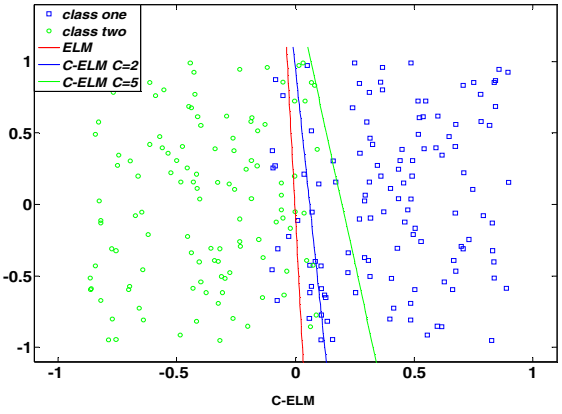


Fig. 1. The effect of the Cost value on decision hyperplane of the C-ELM algorithm

Grid search is the simplest way to find optimal model parameters. A grid structure lends itself naturally t search for discrete parameters while continuous parameters must be discredited, typically by setting a parameter range. Concretely, For an M-labels classification application, we parameterize the cost matrix $C_j, C_j \in [C_{j1}, ...C_{jM}]$ as the CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$, if M=2, the cost matrix (C^+, C^-) as the CP (c^+, c^-), $c^+ \geq c^- > 0$. A grid search of CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$ is conducted in seek of the optimal result. In binary classifier, the positive samples c^+ is searched in a wide range while the negative ones c^- is 1. Nevertheless, in multiclass classifier, c_j constitutes an M dimension grid. An $M \times M$ grid is searched in the cycle. After all the parameters in grid training is completed, we search the best CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$.

Given a parameter c^* (Cost Parameter) and the training set X_{tr}, we can learn the decision model $f(x, c^*, X_{tr}) \in H$ (H is the space of all candidate decision models) according to each of the two algorithms proposed in section 4.1. The corresponding total cost in test set X_{te} is depicted as

$$L(C_m, P_{c^*}(m)) = \frac{1}{M}\sum_{m=1}^{M} C_m P_{c^*}(m), \tag{13}$$

where $P_{c^*}(m), (m = 1, 2, ...M)$ is called the classification accuracy of every classes with Cost Parameter's decision model.

The $f^*(x)$ and c^* can be determined by

$$[f^*(x), c^*, C_m] = \underset{f \in F, c \in [0, c_{\max}]}{\arg\min} L(C_m, P_{c^*}(m)), \tag{14}$$

Therefore, by introducing the CP into the ELM-based cost-sensitive methods with the cost matrix proposed, i.e., C-ELM we can obtain the corresponding the ELM-based two cost-sensitive methods with the CP by the grid-search method .

5.2 The Difference between C-ELM and Other Weighted ELM

In the literature, weighted ELM were proposed by Huang et al. it multiply the error $e_{j,m}$ of the row of sample x_j by the its $w_{j,m}(m = 1, 2, ...M)$, $j = 1, \cdots, N$ .least-square equations can be compactly rewritten as

$$W_R E = W_R(H\beta - Y) = W_R H\beta - W_R Y, \tag{15}$$

where $W_R = w_{j,m}(m = 1, 2, ...M)$, $j = 1, 2 \cdots N$, the H, β and Y is calculated according to (4b). Therefore, by the least-square method, the output weights of weighted ELM can be analytically determined as

$$\hat{\beta}_R = \arg\min_{\beta} \left\| W_R H\beta - W_R Y \right\| = H_R^{\dagger} W_R Y, \tag{16}$$

where $H_R^{\dagger}$ is called the Moore-Penrose generalized inverse of matrix H_R . Moreover, If $H_R^T H_R$ is non-singular, then $H_R^{\dagger} = (H_R^T H_R)^{-1} H_R^T$. While if $H_R H_R^T$ is non-singular, then $H_R^{\dagger} = H_R^T (H_R H_R^T)^{-1}$. Therefore, the decision function is

$$f_R(x) = h(x)(H_R^{\dagger} W_R Y). \tag{17}$$

From decision function, our method is to multiplythe W by Y in the previous , weighted ELM method is to multiplythe W by Y in the back. So, different effect we will achieve.

To verify the theoretical analysis in cost-sensitive ELM ,weighted ELM and ELM , 10 binary datasets are tested in the experiments. Most of the datasets are downloaded online from the UCI repository. In 10 binary datasets, we put some multiclass datasets

into imbalance binary datasets (positive-class examples and negative-class examples) as displayed in Table IV. Datasets contains the number of positive-class examples (Pos.), the number of negative-class examples (Neg.), and the total number of examples (Total), total number of classes, and the class selected as the positive class for each dataset. These datasets represent a whole variety of domains, complexities, and. we set Actual Cost (C^+, C^-) as $(5,1)$.

In training process, different CP lead to different decision models, and different decision models result in different total cost in test datasets. A grid search of CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$ on {1, 2, 3, 4,…, 9 , 10} is conducted in seek of the optimal result. In binary classifier, the positive samples c^+ is searched in a wide range {1, 2, 3, 4, … , 9 , 10}while the negative ones c^- is 1. The two corresponding algorithms with CP increase c^+ from 1 to $c_{\max} = 10$ according to step length of 1. The C-ELM algorithm involves a grid search in which the CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$ can change within the overall search space.

After all the parameters in grid training is completed, we search the best CP $c_j, c_j \in [c_{j1}, ...c_{jM}]$. In the two corresponding algorithms, the 5-fold cross validation is used to determine the parameters in the proposed two ELM-based cost-sensitive algorithms and the ELM. The grid-search method can search Cost Parameter that minimizes total cost.

It can be seen that the C-ELM has better performance than ELM does in term of total cost, and that another algorithms have the nearly equivalent total cost, even

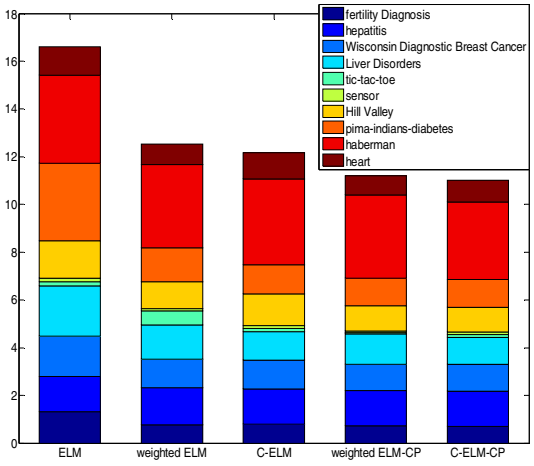


Fig. 2. The performance comparison of the proposed algorithms and ELM in term of total cost on the binary ten benchmark real world datasets

though the C-ELM plays the best. We can conclude that, at least from our experimental results, all the proposed methods are effective in cost-sensitive classification with the unequal error cost. Because the misclassification costs of all of them are apparently less than that of the ELM. Furthermore, the weighted ELM-CP, C-ELM-CP algorithms with the CP have better performance than the corresponding two algorithms with the AC. The results are presented in Fig 2.

6 Conclusion

This paper proposes cost-sensitive ELM based on original ELM for binary and multiclass classification tasks. Comparable or better generalization performance is achievable compared to the conventional machine learning techniques. We focus on the classification, and the class-dependent cost is considered. We first propose cost-sensitive versions of ELM to deal with the classification problems with unequal misclassification cost, in which the cost matrix of misclassifying each class samples is integrated into the ELM algorithm or its decision model. This paper uses the grid-search method to search Cost Parameter. Furthermore, by integrating the CP instead of the cost matrix into the ELM, the C-ELM algorithms are proposed to further improve the performance of the ELM in terms of total cost. We use one artificial datasets and ten real-world binary/multi benchmark datasets to validate the proposed method. In general, the proposed methods have better performance than the ELM and other weighted ELM in terms of total cost.

It would be interesting to extend the work in this paper to other cost-sensitive classification. In addition, the sample-dependent cost can be considered in future work.

Acknowledgements. This work is partly supported by the National Natural Science Foundation of China (60905034) and Natural Science Foundation of Zhejiang province (Y1080950 and Y1080918) and Project of Zhejiang Provincial Department of Education (Y201121959)

References

1. Klaus, N., Jochen, J.S.: Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity. Neurocomputing 102, 23–32 (2013)
2. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing 70, 489–501 (2006)
3. Huang, G.-B., Chen, L.: Convex incremental extreme learning machine. Neurocomputing 70, 3056–3062 (2007)
4. Huang, G.-B., Chen, L.: Enhanced random search based incremental extreme learning machine. Neurocomputing 70, 3460–3468 (2008)
5. Huang, G.-B., Wang, D.-H., Lan, Y.: Extreme learning machine: a survey. Int. J. Mach. Learn. & Cyber. (2), 107–122 (2011)

6. Zong, W.W., Huang, G.B., Chen, L.: Weighted extreme learning machine for imbalance learning. Neurocomputing 101(4), 229–242 (2013)
7. Huang, G.-B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multi-class classification. IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics (2011)
8. Lan, Y., Soh, Y.-C., Huang, G.-B.: Two-stage extreme learning machine for regression. Neurocomputing 73, 223–233 (2010)
9. Feng, G., Huang, G.-B., Lin, Q., Gay, R.: Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans. Neural Netw. 20, 1352–1357 (2009)
10. Huang, G.-B., Chen, L., Siew, C.-K.: Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71, 576–583 (2008)
11. Deng, W.-Y., Zheng, Q.-L., Chen, L.: Regularized extreme learning machine. In: IEEE Symposium on Computational Intelligence and Data Mining, vol. (2), pp. 389–395 (2009)
12. Miche, Y., Sorjamaa, A., Lendasse, A.: OP-ELM: theory, experiments and a toolbox. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 145–154. Springer, Heidelberg (2008)
13. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: optimally pruned extreme learning machine. IEEE Trans. Syst. Man Cybern. Part B Cybern. 21, 158–162 (2010)
14. Zhu, Q.-Y., Qin, A.-K., Suganthan, P.-N., Huang, G.-B.: Evolutionary extreme learning machine. Pattern Recognition 38, 1759–1763 (2005)
15. Liu, N., Han, W.: Ensemble based extreme learning machine. IEEE Singal Processing Letters 17, 754–757 (2010)
16. Qu, Y.-P., Shang, C.-J., Wu, W., Shen, Q.: Evolutionary fuzzy extreme learning machine for mammographic risk analysis. International Journal of Fuzzy Syetems 13, 282–291 (2011)
17. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. In: Proceedings of the 21st National Conference on Artificial Intelligence, Boston, MA, pp. 567–572 (2006)
18. Zhou, Z.H., Liu, X.Y.: The influence of class imbalance on cost-sensitive learning: An empirical study. In: Proceedings of the 6th IEEE International Conference on Data Mining, pp. 970–974 (2006)

Multi-Objective Optimization for Overlapping Community Detection

Jingfei Du, Jianyang Lai, and Chuan Shi*

Beijing University of Posts and Telecommunications, Beijing, China
du1342157416@gmail.com, 330939204@qq.com, shichuan@bupt.edu.cn

Abstract. Recently, community detection in complex networks has attracted more and more attentions. However, real networks usually have number of overlapping communities. Many overlapping community detection algorithms have been developed. These methods usually consider the overlapping community detection as a single-objective optimization problem. This paper regards it as a multi-objective optimization problem and proposes a Multi-Objective evolutionary algorithm for Overlapping Community Detection (MOOCD). This algorithm simultaneously optimize two objective functions to get proper community partitions. Experiments on artificial and real networks illustrate the effectiveness of MOOCD.

Keywords: Complex network, overlapping community detection, multi-objective optimization, evolutionary algorithm.

1 Introduction

In recent years, there is a surge on community detection in complex networks. The main reason lies in that communities play special roles in the structure-function relationship, and thus detecting communities (or modules) can be a way to identify substructures which could correspond to important functions. Generally, communities are groups of nodes that are densely interconnected but only sparely connected with the rest of the network [1][2]. For example, on an online shopping site, users in the same community usually have the same taste in choosing similar goods. However, recent study shows that real networks usually have number of overlapping communities [21]. That is, some nodes in networks exist in multiple communities. It is reasonable in real world, since objects often have multiple roles. For example, a professor collaborates with researchers in different fields; a person has his family group as well as friends group at the same time, etc. So, in overlapping community detection, these objects should be divided into multiple groups.

Up till now, many overlapping community detection algorithms have been developed [11][13][14][20], which can be roughly classified as "node-based" or "link-based" methods. The node-based methods classify nodes of the network directly

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 489–500, 2013.

[20]. The link-based methods cluster the edges of network, and then map the final link communities to node communities by simply gathering nodes incident to all edges within each link communities [11]. The contemporary methods all consider the overlapping community detection as a single-objective optimization problem. That is, the overlapping community detection corresponds to discover a community structure that is optimal on one single-objective function. However, these single-objective algorithms may confine the solution to a particular community structure property because of only considering one objective function. When the optimization objective is inappropriate, these algorithms may fail. Moreover, the overlapping community structure can be evaluated from multiple criteria, which can comprehensively measure the quality of overlapping communities. Although multi-objective optimization has been applied for community detection [17][19], it has not been exploited for overlapping community detection.

In this paper, we first study the multi-objective optimization for overlapping community detection and propose a Multi-Objective evolutionary algorithm for Overlapping Community Detection (MOOCD). The algorithm employs a well-known multi-objective optimization framework for numerical optimization (PESA-II) [22], and uses two conflict objective functions. In addition, the effective genetic representation, operators and model selection strategies are designed. Experiments on typical artificial networks show MOOCD not only accurately detects overlapping communities but also comprehensively reveals community structures. Moreover, experiments on three real networks illustrate that MOOCD discovers more balanceable overlapping communities compared to other well-established algorithms.

2 Related Work

In this section, we will introduce the most related work, including community detection, overlapping community detection , and multi-objective optimization for community detection.

Community detection is crucial for analyzing structures of social networks. There are lots of algorithms aiming at finding proper community partition.One of the most known algorithms proposed so far is the Girvan-Newman (GN) algorithm that introduces a divisive method by iteratively cutting the edge with the greatest betweennes value [3]. Some improved algorithms have been proposed [23][24]. These algorithm are based on a foundational measure criterion of community, modularity, proposed by Newman [3].

Recently,some studies show that real networks usually have number of overlapping communities [21]. Many algorithms have been proposed to detect overlapping communities in complex networks, such as CPM [11], GA-NET+ [13], GaoCD [14], etc. CPM is the most widely used, but its coverage largely depends on the feature of network. GA-NET+, developed by Pizzuti, is the first algorithm that adopts genetic algorithm to detect overlapping communities. However, GA-NET+ costs so much computation in transformation between line graph and node. GaoCD is also a genetic algorithm. But the difference is that GaoCD is

a link-based algorithm. Besides these algorithms, some people extend conventional disjointed community detection criterions to overlapping ones. For example, Shen[15] introduced a practical extended modularity for finding overlapping communities. And Wang[16] also extended modularity Q and proposed an efficient method for adjusting classical algorithms to match the requirement for discovering overlapping communities.

However, because the definition of community is multi-objective, the community detection problem is multi-objective. And the conventional single-objective community detection methods have several crucial disadvantages. Therefore, there are some researchers who have been aware of the multi-objective community detection. For instance, Gong [17] solves the community detection by maximizing the density of internal degrees, and minimizing the density of external degrees simultaneously. Besides, Gong [18] provides a novel multi-objective immune algorithm to solve the community detection problem in dynamic networks. And Shi[19] formulated a multi-objective framework for community detection and proposes a multi-objective evolutionary algorithm for finding efficient solutions under the framework. However, there is few work applies multi-objective community detection methods to find overlapping community partitions.

3 Multi-Objective Evolutionary Algorithm for Overlapping Community Detection

In this section, we will describe the Multi-Objective algorithm for Overlapping Community Detection (MOOCD) in detail, which includes the algorithm framework, objective function, genetic representation, genetic operators and multi-objective model selection method.

3.1 Framework of the Algorithm

This paper applies the evolutionary algorithm (EA) to solve the multi-objective optimization problem. It can simultaneously generate a set of candidate solutions. The framework of MOOCD is described in Algorithm 1.

The framework of MOOCD is based on an existing multi-objective evolutionary algorithm: PESA-II [22]. Different from standard evolutionary algorithms, PESA-II follows standard principles of an EA with the difference that it maintains two populations of solutions: internal population and external population. External population contains non-dominated set, or Pareto front for each updating. A solution dominates other solutions if all objective functions of this solution are superior to other solutions. A solution is said to be Pareto optimal if and only if there is no other solution dominating it. Selection occurs at the interface between the two.

Algorithm 1 randomly generates genes and updates external population at first. At every iteration, internal population is filled with genes selected from external population. New genes are generated based on internal population through genetic operators. And external population is updated by the new genes. After

several iterations, model selection method is used to select a single solution from external population.

3.2 Objective Functions

As we said in Section 2, it is a good choice to use multiple objective functions to solve the drawbacks of the single-objective community detection algorithms. However, it is also a challenge to choose the objective functions. Different objective functions can reflect different characters of partitions. So ideal objective functions had better contain intrinsic conflicts and thus the optimal community partitions can be obtained through the trade-off of multiple objectives. Therefore, the following two objective functions are selected in this paper.

One of the two objective functions is partition density D, which is raised by Ahn [20]. The partition density D is a kind of link community evaluate function whose mathematical definition is as following.

$$D = \frac{2}{M} \sum_{c} m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{1}$$

Define $P = \{P_1, \ldots, P_c\}$ as a partition of the network's links into C subsets. $m_c = |P_c|$ is the number of links in subset c. $n_c = |U_{e_{ij}} \in P_c\{i, j\}|$ represents the number of nodes incident to links in subset c. D_c refers to the link density of subset c. The intuitive meaning of D is the link density within the community. As we said above, partition density is a link based function and it is also appropriate when it is applied to evaluation overlapping community partition.

The other objective function is extended modularity which is proposed by Shen [15]. This objective function is extended from modularity which is used by many community detection methods. Traditional modularity measures the number of within-community edges, relative to a null model of a random graph with the same degree distribution. But we can say that traditional modularity definition cannot be applied to overlapping community detection directly. To adopt Q to overlapping community detection problem, Shen modified the traditional modularity we mentioned above as follow.

$$Q_{OL} = \frac{1}{2m} \sum_{k=1}^{c} \sum_{i,j \in C_k} \frac{1}{O_i O_j} (A_{ij} - \frac{k_i k_j}{2m}) \tag{2}$$

where m is the total number of links in the network, k_i and k_j are the degrees of nodes i and j respectively, A_{ij} are the terms of the adjacency matrix of the network. O_i and O_j are the numbers of communities to which nodes i and j belong respectively.

The two objective functions chosen in this paper are described above. Besides, we can find out through our experiments that partition density tends to find small communities. On the other hand, the modularity optimization may come across the resolution limit problem[25]. From this problem, we can find that modularity can lead the optimization algorithms to large community partitions.

Algorithm 1. Framework of MOOCD

Require:
The set of the internal population, ip_{size};
The set of the external population, ep_{size};
The probability of mutation, p_m;
The probability of crossover, p_c;
The running generation, $gens$;
Ensure:
The final population, P;
$P_{in} = \phi$, $P_{ex} = \phi$
for each i in 1 to ep_{size} do **do**
  g_i = generate_gene()
  calculate_functions(g_i)
  $P_{ex} = P_{ex} \cup \{g_i\}$
end for
for each t in 1 to $gens$ **do**
  $P_{in} = \phi$, $i = 0$, $i = \frac{ip_{size}}{2}$
  in_select(P_{ex}, P_{in}, i)
  while $i < ip_{size}$ **do**
    randomly select two individuals (g_j and g_k) from P_{in}
    generate random value $r \in [0, 1]$
    if $r < p_c$ **then**
      g'_j, g'_k =crossover(g_j, g_k)
    else
      g'_j = mutate(g_j)
      g'_k = mutate(g_k)
    end if
    $i = i + 2$
    calculate_functions(g'_j);$P_{in} = P_{in} \cup \{g'_j\}$
    calculate_functions(g'_k);$P_{in} = P_{in} \cup \{g'_k\}$
  end while
  ex_select($P_{ex}, P_{in}, ip_{size}, ep_{size}$)
end for
P = model_selection(P_{ex})
return P
generate() //initialize individual i according to the genetic representation.
calculate_function(g_i) //evaluate the objective functions of g_i.
ex_select($P_{ex}, P_{in}, ip_{size}, ep_{size}$) //update EP(maximum size is epsize).
crossover(g_i, g_j), *mutate*(g_i) //crossover and mutation genetic operator

These findings reflect the intrinsic conflict between the two. And the experiments in section 4 shows that the algorithm using these functions can find community partitions with different characters.

3.3 Genetic Representation and Operators

In this section, we describe two parts of the algorithm which are encoding and decoding as well as mutation and crossover in detail.

Genetic Encoding and Decoding. To apply genetic algorithm to our problem, we need to transfer the community partitions into some forms which can execute genetic operations. To satisfy the requirement of overlapping community detection, we choose link-based genotype to represent solutions. In this representation method, links are clustered into different partitions. It is possible for nodes that belong to two or more communities. As for the implement of this method, we use the strategy provided by Cai [14].

In this link-based representation, an individual g of the population consists of m genes $\{g_0, g_1, \ldots, g_i, \ldots, g_{m-1}\}$, where $i \in \{0, \ldots, m-1\}$ is the identifier of edges, m is the number of edges, and each g_i can take one of the adjacent edges of edge i. As Fig. 1 shows

Position	0	1	2	3	4	5	6	7	8	9	10
Genotype	1	2	0	4	5	6	7	8	9	10	3

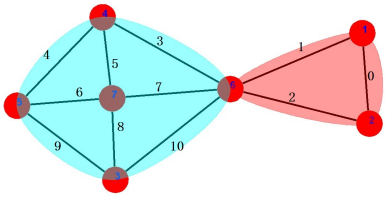


Fig. 1. Genetic Encoding and Decoding

The decoding phase transfers genotype to partition, which consists of link communities. Gene g_i of the genotype and it's value j is interpreted that edge i and edge j have one node in common, and should be classified to same component. In the decoding phase, all components of edges are found, and all edges with in the component constitute a link community.

Genetic Mutation and Crossover. To implement genetic algorithm, we need to confirm some necessary operators such as mutation and crossover. To describe our mutation and crossover strategies, we suppose there are two solutions which are represented through the method above as $g1$ and $g2$. In the crossover operation, we randomly generate a value i. And then, the two genotypes exchange their genes whose positions are i. As for the mutation operation, one random value j is generated. And the jth gene of a certain genotype g is replaced by another value we generate randomly. Fig. 2 show these operations in detail.

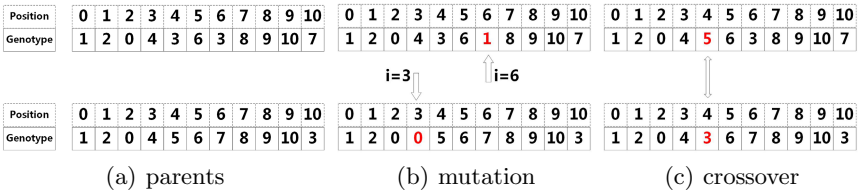


Fig. 2. Genetic Mutation and Crossover

3.4 Model Selection

When the genetic iterations finish, MOOCD returns its external population which is a set of Pareto optimal solutions. And it's time for the decision makers to choose one solution from them. However, we provide an automated strategy to select a more reasonable result. There are many methods that can identify one promising solution in the candidate set. And the principle of some of these model selection methods is to make use of new objective functions to find out a proper solution. In this paper, we use another strategy called Max-Min Distance strategy. The principle of this method is to find a solution which deviates from the random solutions most. And the concrete procedure of it is as follows.

Before the procedure, the method executes MOOCD on some random networks with the same scale. The Pareto front of their solutions are called random Pareto front compared with the real Pareto front.

Firstly, the distance between a solution in the real Pareto front and one in the random Pareto front is defined in

$$dis(C, C') = \sqrt{((intra(C) - intra(C'))^2 + ((intra(C) - intra(C'))^2)}$$

where C and C' represent the solutions in the real and random Pareto fronts, respectively. Then the deviation of a solution in the real Pareto front from the whole random Pareto front is quantified by the minimum distance between this solution and any solutions in the random Pareto front. the deviation is defined in

$$dev(C, CF) = \min\{dis(C, C')|C' \in CF\}$$

where CF represents the random Pareto front. Finally we select the solution in the real Pareto front with the maximum deviation. The model selection process is formulated in

$$S_{max-min} = \arg\max_{C \in SF}\{dev(C, CF)\}$$

where SF represents the real Pareto front.

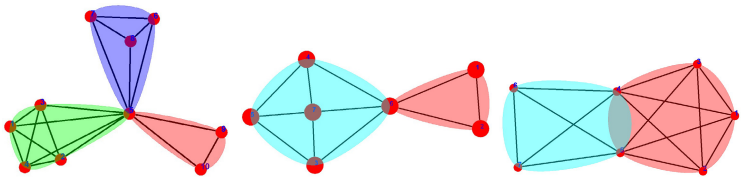
In this section, we provided the framework of MOOCD and described some crucial aspects of our algorithms. And we will evaluate the effectiveness of this method based on artificial networks as well as real networks in the next section. Besides, some other overlapping community detection algorithms are chosen to compare with MOOCD in the experiments.

4 Experiments

This section will validate the effectiveness of MOOCD through experiments on artificial and real networks. The artificial netwrk experiments will illustrate the advantages of multiple solutions returned by MOOCD, and the real network experiments will validate the quality of the solution provided by the model selection method. The experiments are carried out on a 2.2GHz and 2G RAM computer running Windows 7.

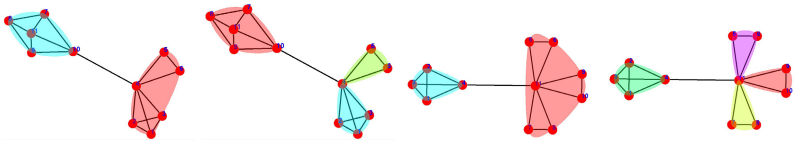
4.1 Experiments on Artificial Networks

To explore the character and ability of MOOCD, we create 5 small typical artificial networks. In the artificial network experiment, we won't use the model selection methods like max-min distance method to choose a single result. Rather, we will represent all the candidate results in Pareto set provided by MOOCD. And these results are shown in Fig. 3 and Fig.4.



(a) N1 (b) N2 (c) N3

Fig. 3. The Community Partition Results of the Artificial Network N1, N2 and N3



(a) N4-1 (b) N4-2 (c) N5-1 (d) N5-2

Fig. 4. The Community Partition Results of the Artificial Network N4 and N5

As we can see in the results, for the networks with single correct community partition (N1,N2,N3), MOOCD can find the correct ovelapping community partitions. More importantly, all the candidate results of MOOCD are meaningful for the networks with multiple community partitions(N4,N5). At the meantime, MOOCD can simultaneously find community partitions in different sizes. This matches our purpose to choose the two objective functions. And we can say that

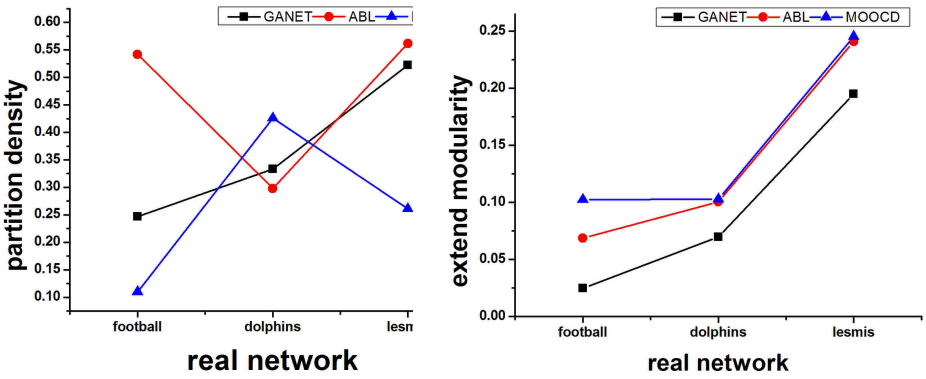
MOOCD has the ability to find different types of communities like overlapping communities and disjoint communities. This conclusion matches our analysis and assumption in Section 3.

4.2 Experiment on Real Network

In this section, we compare MOOCD with ABL and GA_NET on real networks. We execute these 3 algorithm on 3 real networks and calculate D as well as extend Q_{OL} of the partition results. After that, we calculate the number of communities, the size distribution of communities and the average size of communities of the partition results. At last, an intuitive view of partitions on Dolphin found by MOOCD will be posted. Here we choose 3 networks as described in the Table 1 to execute the algorithms. The density and extended modularity are shown in Fig.5.

Table 1. Real Networks Attributes

	dolphins	football	lemis
Nodes	62	115	77
Edges	159	613	254

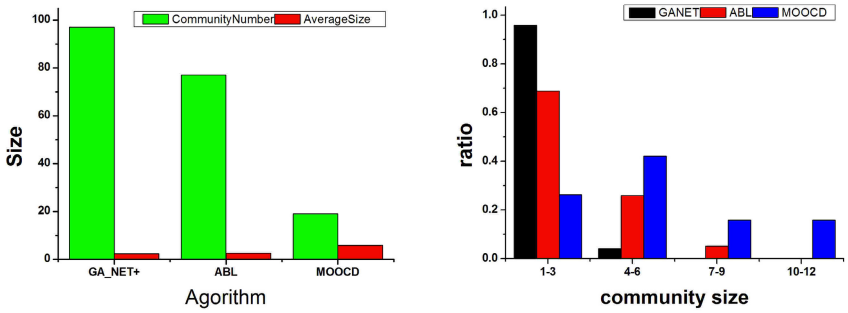


(a) Partition density D (b) Extend modularity Q_{OL}

Fig. 5. The Experiment Result of 3 Methods in Real Networks

The modularity Q_{OL} of our method is much better than that of other methods. Though the partition density of our method is lower than the partition density of other methods in some network, we find a more appropriate community partition through the trade-off between partition density D and extend modularity Q_{OL}. However, these results are not enough to demonstrate the ability and superiority

of MOOCD. Additionally, to evaluate the rationality of the result of MOOCD, in the results of the three methods, we calculate the number of communities, the size distribution of communities and the average size of communities. The results of dolphin network are shown if Fig. 6.



(a) Community Number and Average Size

(b) Community Size Distribution

Fig. 6. Statistic Information for the Partition Results of Dolphin Network

As we can see in this figure, ABL and GA_NET both tend to find small communities and they find too many communities. Noticing that the objective function of GA_NET is community density, we can find community density D can lead the algorithms to find tiny communities. This conclusion demonstrates what we described in Section 3. This doesn't match the real situation. On the other hand, MOOCD can find bigger communities and the size distribution of communities is more balanced.

And then, we show an intuitive view of partition on Dolphin found by MOOCD in Fig.7 and analyze this partition. This figure shows the partitions found by MOOCD in dolphin network. Dolphin network is a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound,

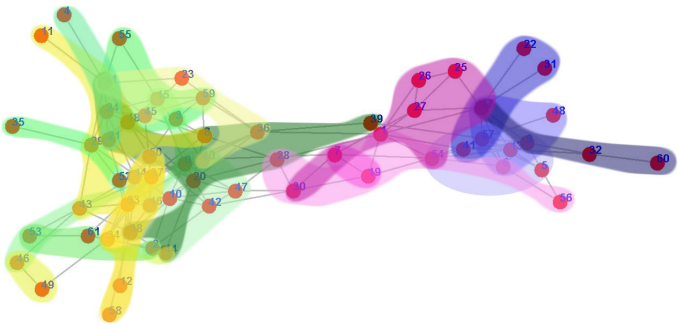


Fig. 7. The Partition of Dolphin Network

New Zealand. The network fell into two parts because of SN100(node 36). In the partition result of MOOCD, node 36 is belonging to many communities. In other word, many communities overlap with each other on the node 36. Removing it makes many communities disjoint with each other, which then splits the networks.

5 Conclusion and Future Work

In this paper, we propose an evolutionary algorithm for multi-objective overlapping community detection. This algorithm uses two classical community partition evaluation functions as objective functions. These objective functions reflect different characters of community structures and make our algorithms have some interesting abilities. The experiments show that our method works well on finding overlapping communities. Besides, MOOCD can simultaneously find different types of community partitions.

In the future, we will try some other interesting objective functions to extend MOOCD. At the meantime, we will apply more than two objective functions to this algorithm. Furthermore, we are going to use the ideas and strategies of this method to solve other problems like dynamic community detection.

Acknowledgement. It is supported by the National Natural Science Foundation of China (No. 61375058, 60905025, 61074128, 71231002). This work is also supported by the National Basic Research Program of China (2013CB329603) and the Fundamental Research Funds for the Central Universities.

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Physics Report 424(4-5), 175–308 (2006)
2. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature 433, 895–900 (2005)
3. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physics Review E 69(026113) (2004)
4. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences 101(9), 2658–2663 (2004)
5. Pothen, A., Sinmon, H., Liou, K.P.: Partitioning sparse matrices with eigenvectors of graphs. SIAM Journal on Matrix Analysis and Applications (11), 430–452 (1990)
6. Kannan, R., Vempala, S., Vetta, A.: On clusterings: good, bad and spectral. Journal of the ACM 51(3), 497–515 (2004)
7. Pizzuti, C.: GA-net: A genetic algorithm for community detection in social networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
8. Shi, C., Yan, Z.Y., Wang, Y., Cai, Y.N., Wu, B.: A genetic algorithm for detecting communities in large-scale complex networks. Advance in Complex System 13(1), 3–17 (2010)

9. Tasgin, M., Bingol, H.: Community detection in complex networks using genetic algorithm. arXiv:cond-mat/0604419 (2006)
10. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review E 72(2), 027104 (2005)
11. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
12. Gregory, S.: An Algorithm to Find Overlapping Community Structure in Networks. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 91–102. Springer, Heidelberg (2007)
13. Pizzuti, C.: Overlapping Community Detection in Complex Networks. ACM (2009)
14. Cai, Y., Shi, C., Dong, Y., Ke, Q., Wu, B.: A Novel Genetic Algorithm for Overlapping Community Detection. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS (LNAI), vol. 7120, pp. 97–108. Springer, Heidelberg (2011)
15. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A 388, 1706–1712 (2009)
16. Xiaohua, W., Licheng, J., Jianshe, W.: Adjusting from disjoint to overlapping community detection of complex networks. Physica A 388, 5045–5056 (2009)
17. Maoguo, G., Lijia, M., Qingfu, Z., Licheng, J.: Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A 391, 4050–4060 (2012)
18. Maoguo, G.: Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. IEEE Transactions on Evolutionary Computation (2013), doi:10.1109/TEVC.2013.2260862
19. Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. Applied Soft Computing 12(2), 850–859 (2012)
20. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466, 761–764 (2010)
21. Palla, G., Dernyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community strcture of complex networks in nature and society. Nature 435, 814 (2005)
22. Corne, D.W., Jerram, N.R., Knwoles, J.D., Oates, M.J.: PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001) (2001)
23. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. PNAS 101(9), 2658–2663 (2004)
24. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E 70(6), 06611 (2004)
25. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences 104(1), 36–41 (2007)

Endmember Extraction by Exemplar Finder

Yi Guo[1,⋆], Junbin Gao[2], and Yanfeng Sun[3]

[1] CSIRO Division of Computational Informatics,
North Ryde, NSW 1670, Australia
yi.guo@csiro.au
[2] School of Computing and Mathematics,
Charles Sturt University, Bathurst, NSW 2795, Australia
jbgao@csu.edu.au
[3] Beijing Municipal Key Lab of Multimedia and Intelligent Software Technology,
Beijing University of Technology, Beijing 100124, China
yfsun@bjut.edu.cn

Abstract. We propose a novel method called exemplar finder (EF) for spectral data endmember extraction problem, which is also known as blind unmixing in remote sensing community. Exemplar finder is based on data self reconstruction assuming that the bases (endmembers) generating the data exist in the given data set. The bases selection is fulfilled by minimising a $l2/l1$ norm on the reconstruction coefficients, which eliminates or suppresses irrelevant weights from non-exemplar samples. As a result, it is able to identify endmembers automatically. This algorithm can be further extended, for example, using different error structures and including rank operator. We test this method on semi-simulated hyperspectral data where ground truth is available. Exemplar finder successfully identifies endmembers, which is far better than some existing methods, especially when signal to noise ratio is high.

1 Introduction

The method proposed in this paper arises from a particular application in remote sensing called blind unmixing [1]. The data acquired by the sensors mounted on unmanned aerial vehicles or satellites are spectra in which different materials usually have distinct signatures. One of the central tasks of analysing these spectra is to recover the constituents of the scene that a device observes for earth observation purpose. Denote a spectral sample by $\mathbf{x}_i \in \mathbb{R}^D$ where D is the number of spectral bands (wavelengths), say 321 bands from 6um to 14um for a typical thermal infrared spectrum. It is possible that $\mathbf{x}_i$ is a spectrum of a pure material, for example, a pure Chlorite sample. However, it is also likely that $\mathbf{x}_i$ is composed by several things mixed together. We normally work on a simplified model where we can approximate the mixing process by a so-called linear mixing model [2] as follows

$$\mathbf{x}_i = \sum_{j=1}^{M} \alpha_{ij}\mathbf{p}_j + \boldsymbol{\epsilon}_i, \text{ s.t. } \alpha_{ij} \geq 0,\ j = 1, \ldots, M \text{ and } \sum_{j=1}^{M} \alpha_{ij} = 1, \tag{1}$$

⋆ The author to whom all the correspondences should be addressed.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 501–512, 2013.

where $\mathbf{p}_j$ is the spectrum of the jth known material, α_{ij} is the corresponding coefficient or proportion of $\mathbf{p}_j$ and $\boldsymbol{\epsilon}_i$ is the error. We assume that $\mathbf{x}_i$ consists of M materials and M needs to be determined. The nonnegative and sum to one conditions in (1) reflects the intuitive interpretation of α_{ij}'s being proportions. Apparently, we need to know the spectra of pure materials beforehand, so that we can then "unmix" a given spectrum $\mathbf{x}_i$ to M materials that are most likely to be present. This process is often called linear unmixing [3].

A spectral library of pure spectra is essential in linear unmixing. A lot of efforts have been dedicated to building a reliable library in the last decades for a variety of devices. However, there is no such a universal library that is suitable for every spectrometer in the world simply because different sensors have different characteristics. So it is still often the case that the library is missing. Blind unmixing [4] then becomes useful when little information is available about the materials in the scene. It deals with the case where $\mathbf{p}_j$'s are unknown.

Clearly this is an ill-posed problem because all variables in (1) but $\mathbf{x}_i$ need to be inferred from the data. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ be the matrix of the given spectral data set of a scene, and $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_P] \in \mathbb{R}^{D\times P}$ the matrix of all P endmembers and $P \ll N$ in general. Based on the linear mixing model (1), we have

$$\mathbf{X} = \mathbf{PA}, \text{ s.t. } \mathbf{A} \geq 0 \text{ and } \mathbf{1}_P^\top \mathbf{A} = \mathbf{1}_N, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{P\times N}$ is the proportion matrix stacking all α_{ij}'s and $\mathbf{1}_P$ is the vector of all 1 of size P. In the setting of blind unmixing, both $\mathbf{P}$ and $\mathbf{A}$ need to be estimated. Interestingly, it can be seen as a matrix decomposition problem. This immediately brings many matrix decomposition methods into consideration, for example PCA [5] and nonnegative matrix factorization [6,7]. Indeed these methods play important roles in blind unmixing. It can also be regarded as a dimensionality reduction problem [8], which tries to find a low dimensional representation of the data, i.e. $\mathbf{A}$ and the bases $\mathbf{P}$. This interpretation is yet to be pursued. Nevertheless, the most challenging part is the lack of information about $\mathbf{P}$ so that we have to make some reasonable assumptions.

Quite a few methods have been proposed for blind unmixing, for example, N-FINDR [9], endmember bundles [10], Iterative Constrained Endmembering (ICE) [4], Minimum Volume Constrained Nonnegative Matrix Factorization [11] just to name a few. As we mentioned earlier, assumptions are essential for blind unmixing. One of the most important one is that the pure materials exist in the scene, i.e. some $\mathbf{x}_i$'s are spectra of pure materials, and others are mixtures of them. An opposite view is that every sample must be mixture of several things. However, as spectral sensors technology advancing so fast, now this is a reasonable assumption and it makes practical sense since the spatial resolution of spectrometers is getting much finer so that it is possible that some samples are indeed pure. Furthermore, if all samples are mixtures, it can be shown that the solution is plagued by indeterminacy, i.e. infinitely many viable solutions, if there is no prior knowledge at all for the endmembers. For these reasons, we consider the case that the assumption of existence of pure materials holds in this paper.

The method proposed in this paper is called Exemplar Finder (EF). Unlike N-FINDR, a typical method exploiting the assumption of presence of pure materials, which is simplex finding based algorithm and therefore computational intensive, EF is based on

data self-reconstruction with data selection embedded. It has a simple convex objective function with a unique solution and therefore easy to solve. The next two sections will introduce this new algorithm and its implementation in detail. We demonstrate its effectiveness in the experimental results section and conclude this paper by a discussion.

2 Exemplar Finder Model

Our proposed Exemplar Finder model has the following form

$$\min_{\mathbf{Z}} \frac{1}{2}||\mathbf{E}||_F^2 + \lambda||\mathbf{Z}||_{2/1}, \text{ s.t. } \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \ \mathbf{Z} \geq 0 \text{ and } \mathbf{1}_N^\top \mathbf{Z} = \mathbf{1}_N, \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times N}$ is the self-reconstruction coefficient matrix, $||\mathbf{M}||_F$ is the Frobenius norm of matrix $\mathbf{M}$ and $||\mathbf{M}||_{2/1}$ is the row dominant $l2/l1$ norm of a matrix defined as $||\mathbf{M}||_{2/1} = \sum_i ||\mathbf{m}^i||_2$, where $\mathbf{m}^i$ is the ith row of matrix $\mathbf{M}$ and $||\mathbf{m}||_2$ is the $l2$ norm of vector $\mathbf{m}$.

The data self-reconstruction with $l2/l1$ minimization carries out the exemplar finding task. Due to the presence of pure materials spectra assumption, we know that the data must be able to be reconstructed by those pure materials with minimum error according to the linear mixing model (1). Therefore, examining the patterns of $\mathbf{Z}$, i.e. different reconstruction solutions, one can find that only the solution using those true pure spectra has minimum number of nonzero or dominant rows and minimum reconstruction error. Based on this observation, we minimize the scale of the reconstruction error and the $l2/l1$ norm of $\mathbf{Z}$. Here we use the group elimination property of $l2/l1$ norm minimisation [12]. In our case, it sets rows to be zero when their influence is minor. This justifies the EF model in (3). The dominant rows with largest $l2$ norms indicate the pure materials in the data set, which we call exemplars in this model. Due to its exemplar identification capability, we call this method Exemplar Finder. This is similar to the idea of subspace learning [13,14] where the data self-reconstruction is exploited to recover subspaces. However, unlike subspace learning, we do not restrict the diagonal elements of $\mathbf{Z}$ to be zero since $l2/l1$ norm minimisation can rule out the trivial solution of $\mathbf{Z}$ being an identity matrix.

The error is measured by least squares, i.e. $||\mathbf{E}||_F^2$. This is equivalent to assuming that the error is from a standard Gaussian distribution. Interestingly, this can be possibly extended further to other measurements, for example, $l1$ norm, $l2/l1$ norm and so on. In those cases, the error structures are different. We use least squares here for its simplicity and reasonably good performance.

Nonnegativity and sum to one constraints are in place complying with the requirement of linear mixing model (1). So matrix $\mathbf{Z}$ should give us the proportions of the pure materials directly when (3) is optimised. However, these constraints bias the $l2/l1$ norm minimisation a little in the sense that it can never eliminate a row entirely. This can be easily solved by refitting after pure materials identification. This will be discussed later.

Other blind unmixing methods have very different models. For example, ICE optimises the following cost function

$$\min_{\mathbf{P},\mathbf{A}} ||\mathbf{E}||_F^2 + \lambda \sum_{i \neq j} ||\mathbf{p}_i - \mathbf{p}_j||_2^2, \text{ s.t. } \mathbf{X} = \mathbf{PA} + \mathbf{E}, \ \mathbf{A} \geq 0, \text{ and } \mathbf{1}_P^\top \mathbf{A} = \mathbf{1}_N. \tag{4}$$

The idea is to find P endmembers that circumscribe most of the data. The optimisation is carried out iteratively alternating optimisation of $\mathbf{P}$ or $\mathbf{A}$ while fixing the other. Apparently the objective function in (4) is not convex meaning that there are many initialisation dependent sub-optima. Nevertheless, ICE is a landmark algorithm for blind unmixing due to its flexibility of accommodating other constraints. Its optimisation scheme is also adopted widely by many variants. For example, the sparse promoted iterated constrained endmember detection (SPICE) [15] introduces a sparse encouraging norm on $\mathbf{A}$ as follows

$$\min_{\mathbf{P},\mathbf{A}} ||\mathbf{E}||_F^2 + \lambda_1 \sum_{i \neq j} ||\mathbf{p}_i - \mathbf{p}_j||_2^2 + \lambda_2 ||\mathbf{A}||_1, \text{ s.t. } \mathbf{X} = \mathbf{P}\mathbf{A} + \mathbf{E}, \ \mathbf{A} \geq 0, \ \mathbf{1}_P^\top \mathbf{A} = \mathbf{1}_N. \tag{5}$$

The argument of SPICE is that most of samples are mixture of several endmembers. This is reflected by the $l1$ norm minimisation on $\mathbf{A}$.

Other structures are imposed on the ICE type of model such as nonnegativity constraint on $\mathbf{P}$, different sparsity encouraging norms on $\mathbf{A}$ and so on. One drawback of these variants inherited from ICE is the non-convexity of the objective function. The initialisation scheme they adopted normally is choosing P samples randomly from data set. However, there is no grantee that the global optimum can be obtained.

3 Implementation

We proceed to implementation of Exemplar Finder model in this section. It is clear that the problem in (3) is convex. Therefore, there will be a unique solution. Here is the summary of the optimisation algorithm for EF. We adopt Nesterov's method [16,17] as the skeleton structure, within which the main update for $\mathbf{Z}$ is carried out by the $l2/l1$ minimisation procedure discussed in [18].

We first eliminate the equality constraint in (3) and write the objective function as

$$\mathcal{L}(\mathbf{Z}) = \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{Z}||_F^2 + \lambda ||\mathbf{Z}||_{2/1}.$$

We take the first order approximation of the quadratic part at a given $\mathbf{S}$ with an additional proximal term as

$$\mathcal{L}_{\gamma,\mathbf{S}}(\mathbf{Z}) = \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{S}||_F^2 + \langle \mathbf{G}, \mathbf{Z} - \mathbf{S} \rangle + \frac{\gamma}{2}||\mathbf{Z} - \mathbf{S}||_F^2 + \lambda ||\mathbf{Z}||_{2/1} \tag{6}$$

where $\mathbf{G} \in \mathbb{R}^{D \times N}$ is the gradient of $||\mathbf{X} - \mathbf{X}\mathbf{Z}||_F^2$ at $\mathbf{S}$ and $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^\top \mathbf{B})$ is the inner product of two matrices. The purpose of including the proximal term in the linearisation is to ensure the estimation is not too far away from the given point $\mathbf{S}$. γ controls the step size in the estimation. We maximize the step size in the algorithm to be discussed later for fast speed while maintaining the necessary accuracy.

Nesterov's method uses two sequences to approach the optimum, $\mathbf{Z}_k$ and $\mathbf{S}_k$, where $\mathbf{S}_k = \mathbf{Z}_k + \beta_k(\mathbf{Z}_k - \mathbf{Z}_{k-1})$. $\mathbf{Z}_k$ is obtained by solving $\min_{\mathbf{Z}} \mathcal{L}_{\gamma,\mathbf{S}_k}(\mathbf{Z})$. The program begins with an initialization of $\mathbf{Z}$, $\mathbf{Z}_0$, and set $\mathbf{Z}_0 = \mathbf{Z}_1$. In cold start case, it is simply a zero matrix. An appropriate value of γ is chosen using the Armijo-Goldstein rule [17].

It works with a special update rule for β_k to achieve an $\mathcal{O}(\frac{1}{k^2})$ convergence rate, which is optimal for first order algorithms. See [17, chapter 2] for details. We sketch the main algorithm in Table 1. The stopping criterion could be specified as the maximum number of iterations and/or a continuous update threshold.

In Step 5, $\arg\min_{\mathbf{Z}} \mathcal{L}_{\gamma,\mathbf{S}_k}(\mathbf{Z})$ can be simplified as

$$\tilde{\mathbf{Z}} = \arg\min_{\mathbf{Z}} \frac{1}{2}||\mathbf{Z}-\mathbf{V}||_2^2 + \lambda||\mathbf{Z}||_{2/1} \tag{7}$$

where $\mathbf{V} = \mathbf{S} - \mathbf{G}/\gamma$. This has a closed form solution [18]. $\mathcal{P}_{\mathcal{C}}(X)$ in step 6 is the Euclidean projection of X to convex set $\mathcal{C}$, meaning that $\mathcal{P}_{\mathcal{C}}(X) = \arg\min_{Y,Y\in\mathcal{C}} ||Y - X||_F^2$. Precisely, we solve the following problem

$$\min_{\mathbf{Z}} ||\mathbf{Z}-\tilde{\mathbf{Z}}||_F^2, \text{ s.t. } \mathbf{Z} \geq 0, \text{ and } \mathbf{1}_N^\top \mathbf{Z} = \mathbf{1}_N, \tag{8}$$

where $\tilde{\mathbf{Z}}$ is the solution to (7). This is a quadratic programming (QP) problem which can be solved by standard QP solvers.

After we obtain the optimal $\mathbf{Z}$, we calculate $l2$ norm of each row of $\mathbf{Z}$, i.e. $||\mathbf{z}^i||_2$, and the identified endmembers are the ones with P largest $l2$ norms. The final algorithm of EF is summarised in Table 2.

Table 1. Main algorithm for EF. $\mathcal{P}_{\mathcal{C}}(X)$ in step 6 is the Euclidean projection of X to the convex set $\mathcal{C}$ corresponding to constraints

Optimize (3) ***via Nesterov's method***
Input: $\mathbf{X}$, λ, $\mathbf{Z}_0$
Output: optimal $\mathbf{Z}$
1. Initialization: $\mathbf{Z}_1 = \mathbf{Z}_0$, $\gamma = 1$, $l_{-1} = l_0 = 1$.
2. for $k = 1$ to ...
3. $\beta_k = \frac{l_{k-2}-1}{l_{k-1}}$, $\mathbf{S}_k = \mathbf{Z}_k + \beta_k(\mathbf{Z}_k - \mathbf{Z}_{k-1})$
4. for $j = 1$ to ...
5. $\tilde{\mathbf{Z}}_{k+1} = \arg\min_{\mathbf{Z}} \mathcal{L}_{\gamma,\mathbf{S}_k}(\mathbf{Z})$
6. $\mathbf{Z}_{k+1} = \mathcal{P}_{\mathcal{C}}(\tilde{\mathbf{Z}}_{k+1})$
7. if $\mathcal{L}(\mathbf{Z}) \leq \mathcal{L}_{\gamma,\mathbf{S}_k}(\mathbf{Z})$ then $\gamma = \max(2\gamma, \frac{||\mathbf{X}(\mathbf{Z}_{k+1}-\mathbf{S}_k)||_F^2}{||\mathbf{Z}_{k+1}-\mathbf{S}_k||_F^2})$
8. else break
9. end if
10. end for
11. $l_k = (1+\sqrt{1+4l_{k-1}^2})/2$
12. if convergence, then stop and output $\mathbf{Z}_{k+1}$ as the solution.
13. end for

Table 2. Complete exemplar finder algorithm. $\mathbf{X}_{\mathcal{S}}$ is the submatrix of $\mathbf{X}$ stacking the columns indexed by $\mathcal{S}$

Exemplar Finder Algorithm
Input: $\mathbf{X}$, λ, $\mathbf{Z}_0$, P (number of endmembers)
Output: P endmembers identified in $\mathbf{X}$
1. Obtain optimal solution to (3) using algorithm in Table 1;
2. Calculate $l2$ norm of each row of $\mathbf{Z}$, $\mathbf{d} = [||\mathbf{z}^1||_2, \ldots, ||\mathbf{z}^N||_2]$;
3. $\mathbf{P} = \mathbf{X}_{\mathcal{S}}$ where $\mathcal{S} = \{$indices of the P largest elements in $\mathbf{d}\}$.

Remark 1 (Exemplar identification). *Normally when minimising the row dominant $l2/l1$ norm of a matrix, one would expect that many rows of the matrix have been set*

to zero when the optimisation completes. However, under the nonnegative and sum to one constraints, the minimisation of $l2/l1$ cannot really annihilate rows completely. It can be observed that if we choose a large value of λ, we end up with a matrix with all elements equal to $1/N$. For this reason we use step 2 and 3 in Table 2 to identify the exemplars instead of detecting zeros rows in $\mathbf{Z}$. This strategy is actually more robust. We will show this in the experimental results section.

Remark 2 (Regularisation paramter). *There is a regularisation parameter λ in EF algorithm. The choice of this parameter usually connects to model selection. However, since we do not detect zero rows in $\mathbf{Z}$ but identify only the P most significant rows in terms of $l2$ norm, this eases the importance of the choice of λ. Throughout the experiments, we simply set it to $0.01N$ and it worked very well. This suggests that we can actually fix it and henceforth no tuning parameter at all in EF algorithm.*

Remark 3 (The number of exemplars). *In most blind unmixing methods, the number of endmembers must be specified. It can be detected by some methods such as virtual dimensionality [19] or dimensionality estimation procedures [20,21]. In EF algorithm, we assume that P is given. However, it is very likely this number can be detected automatically from the pattern of $l2$ norms of the rows of $\mathbf{Z}$. We will show this as well in the experiment section.*

4 Experimental Results

We will show some experimental results to demonstrate the effectiveness of EF algorithm with a comparison with two popular methods, ICE and SPICE, in this section. Note that we have to restrict our experiments on the so-called semi-simulated data set [22] where we know exactly the ground truth, i.e. the true endmembers, so that we can evaluate the effectiveness of the methods quantitatively.

4.1 Experiment Settings

The semi-simulation is based on thermal infrared (TIR) spectral data. We have a library of 120 typical thermal infrared spectra of pure materials with 321 thermal infrared wavelengths (6um - 14um). Background removal was carried out for the library data. Background corresponds to temperature, lighting condition and other aspects of acquisition environment which effect the shape of the spectra. The background removal reduced most variation irrelevant to spectral features. We randomly picked P spectra from the library as our endmembers, and sampled the weights from a uniform distribution between [0,1] and re-scaled to proportions (sum to one) to get 100 simulated noise free spectral data for tests. We obtained noise mean and covariance as in [22] to simulate additive noise so that the simulated spectra are close to reality. We also mixed the true endmember into the data to satisfy the presence of pure materials assumption.

In regard to the EF optimization algorithm, we chose the maximum number of iterations to be 500 and a continuous objective function difference threshold 10^{-6}. Whichever criterion is achieved first stops the optimization. As mentioned earlier, we set $\lambda = 0.01N$ for EF in all experiments. For other methods, we set the recommended values to parameters or searched the optimal values if the default ones were not satisfying.

4.2 Noise Free Cases

Fixed Number of Endmembers. To our best experience in thermal infrared spectra, most of the real spectra from proximal sensing (spectra are taken on ground not from the air) are composed by up to 6 materials. Based on this, we first fix $P = 6$ to test algorithms on noise free data.

Figure 1 (a) and (b) shows the 6 endmembers randomly chosen from the TIR library and 100 simulated spectra including the endmembers. It should be a fairly straightforward case for endmembers extraction. The endmembers found by different methods are plotted in Figure 1 (c) - (e). All these methods worked reasonably well to identify 5 out of 6 endmembers. However, ICE and SPICE missed one with substitutes mixing other features, while EF obtained all true endmembers exactly.

We used the following matching costs to quantify an endmember extraction solution:

$$c_i = \min ||\mathbf{p}_i - \hat{\mathbf{p}}_j||_2^2,\ j = 1, \ldots, P, \tag{9}$$

and

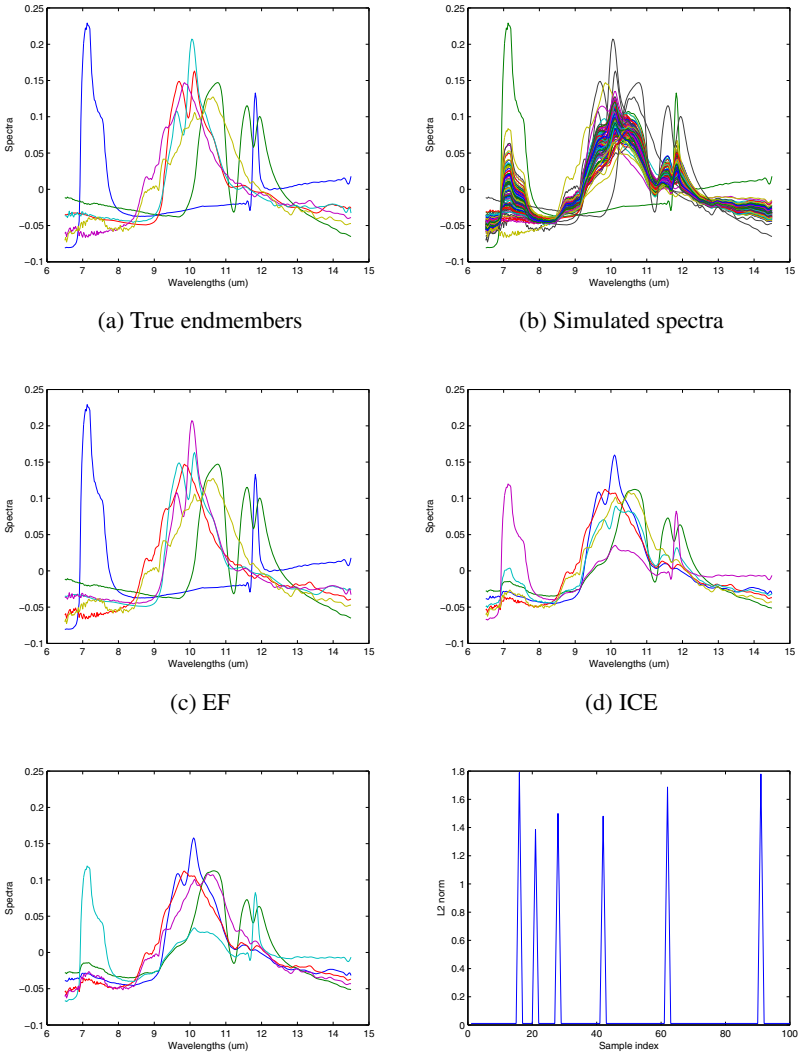
$$C = \frac{1}{\min\{P, \hat{P}\}} \sum_{i}^{P} c_i, \tag{10}$$

where c_i is the matching cost for the ith true endmember $\mathbf{p}_i$, C is the overall matching cost and $\hat{\mathbf{p}}_j$ is the jth estimated endmember by an algorithm ($j = 1, \ldots, \hat{P}$). Since true endmembers and estimated endmember are two sets of data, finding a perfect match between them is a bipartite matching problem. We simply use (9) to find the best match for $\mathbf{p}_i$ in terms of $l2$ distance and the minimum $l2$ distance is the matching cost for $\mathbf{p}_i$. The overall matching cost is the average of individual matching cost. Note that some solutions may not produce specified number of endmembers, for example, SPICE generates less than specified number of endmembers sometimes. This is punished by a smaller denominator in (10). As the name suggests, the smaller the cost, the better the solution.

The detailed matching cost of three methods on this data set is listed in Table 3. It shows clearly that ICE and SPICE have trouble with the first endmember and not so well with the second one. The matching cost evaluation confirms our visual interpretation and reveals a lot more details. Again, Table 3 shows the perfect identification results by EF.

Figure 1 (f) shows the $l2$ norm pattern of the rows of the optimal $\mathbf{Z}$ obtained by EF. The $l2$ norms of the rows of $\mathbf{Z}$ associated with pure materials are much larger than those of the others, which are not exactly zero. So if we threshold the $l2$ norms of the row vectors of $\mathbf{Z}$, we should be able to detect the number of exemplars automatically. As the eigenvalue analysis in PCA, sometimes it may be difficult to choose a clear cutoff line. Human intervention could help in this scenario.

Varying Number of Endmembers. Based on the quantitative measurement presented in last section, we proceed to investigate the effectiveness of EF on the data sets with different number of endmembers. We varied P from 1 to 16 and generated the test data in the way as we mentioned in Section 4.1. For every value of P, we repeated the experiment 10 times to examine how well these methods performed by evaluating the



(a) True endmembers (b) Simulated spectra

(c) EF (d) ICE

(e) SPICE (f) $l2$ norm pattern of $\mathbf{Z}$ by EF

Fig. 1. (a)-(b):6 true endmembers randomly chosen from the TIR library and 100 simulated spectra. (c) - (e): 6 endmembers found by three different methods. (f): $l2$ norm pattern of row vectors of $\mathbf{Z}$ obtained by EF.

Table 3. Matching cost of different methods. The head of the table is the indices of true endmembers.

Algorithms	1	2	3	4	5	6	Overall
EF	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ICE	0.3590	0.1133	0.0311	0.0259	0.0391	0.0130	0.0969
SPICE	0.3623	0.1133	0.0304	0.0285	0.0383	0.0123	0.1170

means and variances of their matching costs. The results are shown as the mean and variance plot in Figure 2. EF is able to identify all the endmembers successfully across all cases. However, the performance of ICE and SPICE deteriorates as the number of endmembers to be detected increases.

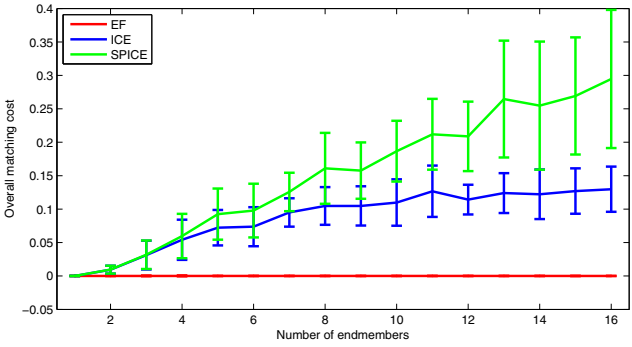


Fig. 2. The means and variances of the matching costs of three methods

4.3 Noisy Spectra

As shown in the previous sections, EF has clear advantage in identifying pure materials when the spectra are clean. However, the acquired spectra from spectrometers have more or less noise in reality. The noise level is often measured by SNR (signal to noise ratio). In this section, we test the methods on controlled noisy data so that the performance can be evaluated precisely.

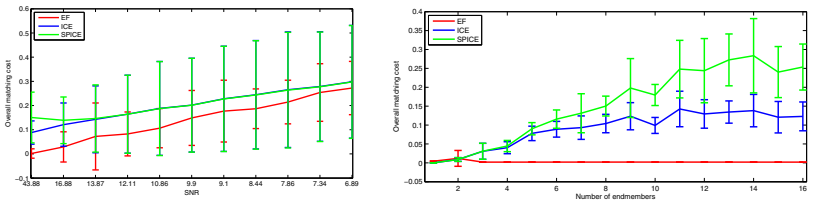
After generated the test data using given number of endmembers, we added Gaussian noise from $\mathcal{N}(\mathbf{m}, \delta\mathbf{\Sigma})$ where $\mathbf{m}$ and $\mathbf{\Sigma}$ are the noise mean and covariance matrix mentioned in Section 4.1, and δ is the scale factor controlling the noise level. We use the overall SNR across all bands of a spectra data set defined as

$$\text{SNR} = -10\log 10(\frac{||\mathbf{S}-\mathbf{T}||_F^2}{||\mathbf{S}||_F^2}),$$

assuming $\mathbf{S}$ is the noise free signal and $\mathbf{T}$ is the noise contaminated signal. This boils down to SNR$= -10\log 10\delta\text{tr}(\mathbf{\Sigma})$ since the noise is additive. The simulated noise level δ varied from 1 to 5001 corresponding to SNR from 44db to 7db. The SNR of a practical spectrometer varies between 30db and 60db. So the noise level of this experiment

covers one third of the range of real devices with the focus on really noisy spectra with SNR down to 7db. We varied the number of endmembers from 1 to 16. To evaluate the stability of different methods, we repeated the experiment 10 times so we can assess the means and variances of different solutions.

Figure 3 (a) plots the overall matching cost of three methods for the cases with 1 to 16 endmembers against different SNR values. It summarises the performance of a method across 10 repeated random experiments. The SNR axis in Figure 3 (a) is not in log scale in order to get equal interval plot. Note that we matched the endmembers extracted by any algorithm to the true clean endmembers and we did not apply any smoothers to extracted endmembers or test spectra. This is very challenging to EF as it only extracts exemplars from given data and therefore the solution endmembers contained the same amount of noise as the data set. These extracted endmembers were matched to clean ground truth so that the matching cost involved noise. Whereas ICE and SPICE have some smoothing effect since the endmembers are calculated instead of selected. Nevertheless, several facts can be observed from the variance plot. First, the performance of all methods drops when SNR is very low. Second, EF has lower matching costs than others constantly across all SNR scenarios, although the detected endmembers were not smoothed. Third, ICE and SPICE are almost the same because their results are smoothed by mean operation and also they are indeed similar in model structure.



(a) Vary SNR (b) SNR=27

Fig. 3. (a) The overall matching cost of different methods across solutions for number of endmembers from 1 to 16. (b) The matching cost of different methods when SNR=27.

We close this section by presenting the results of three methods with SNR equals 27 which is the typical signal quality of a spectrometer. The number of endmembers is from 1 to 16 with 10 repeats for every test. Figure 3 (b) shows the results. Once again, EF has very clear advantage when the spectra quality is good. However, we need to point out that when the number of endmembers P equals 2, the variance of EF results is remarkably large compared with that of EF with other values of P. This phenomenon eludes us and it needs further investigation.

5 Discussion

We proposed exemplar finder algorithm (EF) in this paper for endmember extraction problem. EF model is based on the assumption of existence of pure materials,

i.e. exemplars, in the data set. It takes the form of data self-reconstruction with a row dominant $l2/l1$ minimisation for bases selection. The objective function of EF is convex and therefore has a unique solution. The optimisation is carried out by Nestorove's method to achieve $\mathcal{O}(\frac{1}{k^2})$ convergence. The final exemplar extraction is fulfilled by selecting row vectors of reconstruction matrix $\mathbf{Z}$ with P largest $l2$ norms. We tested EF against other two popular blind unmixing methods called ICE and SPICE using controlled semi-simulated data with and without noise. The experimental results demonstrated that EF is better than those two methods. Although our focus in this paper is endmember extraction, EF is not limited to this application. It can also be applied to problems in image processing, computer vision and other fields where exemplar detection is essential.

Acknowledgements. The work of second author is supported by Australian Research Council (ARC) under grant DP130100364. The work of the fourth author is supported by National Nature Science Foundation of China (NFSC) under grant No. 61370119.

References

1. Bioucas-Dias, J.M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J.: Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE Transactions on Geoscience and Remote Sensing 5(2), 354–379 (2012)
2. Guo, Y., Berman, M.: A comparison between subset selection and L1 regularisation with an application in spectroscopy. Chemometrics and Intelligent Laboratory Systems 118, 127–138 (2012)
3. Berman, M., Bischof, L., Lagerstrom, R., Guo, Y., Huntington, J., Mason, P.: An unmixing algorithm based on a large library of shortwave infrared spectra. Technical Report EP117468, CSIRO Mathematics, Informatics and Statistics (2011)
4. Berman, M., Kiiveri, H., Lagerstrom, R., Ernst, A., Dunne, R., Huntington, J.F.: ICE: A statistical approach to identifying endmembers in hyperspectral images. IEEE Transaction on Geoscience and Remote Sensing 42(10), 2085–2095 (2004)
5. Jolliffe, M.: Principal Component Analysis. Springer, New York (1986)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature 401(6755), 788–791 (1999)
7. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Int. Conf. Adv. Neural Inf. Process. Syst., vol. 13, pp. 556–562 (2001)
8. Guo, Y., Gao, J., Kwan, P.W.: Twin kernel embedding. IEEE Transaction of Pattern Analysis and Machine Intelligence 30(8), 1490–1495 (2008)
9. Winter, M.: Fast autonomous spectral endmember determination in hyperspectral data. In: 13th Int. Conf. Applied Geologic Remote Sensing, pp. 337–344 (1999)
10. Bateson, C.A., Asner, G.P., Wessman, C.A.: Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis. IEEE Transactions on Geoscience and Remote Sensing 38(2), 1083–1094 (2000)
11. Miao, L., Qi, H.: Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. IEEE Transaction on Geoscience and Remote Sensing 45(3), 765–777 (2007)
12. Guo, Y., Gao, J., Li, F.: Dimensionality reduction with dimension selection. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS, vol. 7818, pp. 508–519. Springer, Heidelberg (2013)

13. Vidal, R.: A tutorial on subspace clustering. In: CVPR (2010)
14. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning (2010)
15. Zare, A., Gader, P.: Sparsity promoting iterated constrained endmember detection in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing Letters 4(3), 446–450 (2007)
16. Nemirovski, A.: Efficient Methods in Convex Programming. Lecture Notes (1995)
17. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Applied Optimization, vol. 87. Kluwer Academic Publishers (2004)
18. Liu, J., Moreau-Yosida, J.Y.: Regularization for grouped tree structure learning. In: Int. Conf. Adv. Neural Inf. Process. Syst. (2010)
19. Chang, C.I., Du, Q.: Estimation of number of spectrally distinct signal sources in hyperspectral imagery. IEEE Transaction on Geoscience and Remote Sensing 42(3), 608–619 (2004)
20. Raginsky, M., Lazebnik, S.: Estimation of intrinsic dimensionality using high-rate vector quantization. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems 18, pp. 1105–1112. MIT Press, Cambridge (2006)
21. Carter, K.M., Raich, R., Iii, A.O.H.: On local intrinsic dimension estimation and its applications. IEEE Transactions on Signal Processing 58(2), 650–663 (2010)
22. Guo, Y., Gao, J., Li, F.: Spatial subspace clustering for hyperspectral data segmentation. In: ICDIPC 2013, pp. 181–191 (2013)

EEG-Based User Authentication in Multilevel Security Systems

Tien Pham[1], Wanli Ma[1,2], Dat Tran[1], Phuoc Nguyen[1], and Dinh Phung[1]

[1] Faculty of Education, Science, Technology and Mathematics,
University of Canberra, Australia
[2] Department of Computer Science, University of Houston Downtown, USA
{tien.pham,wanli.ma,dat.tran,phuoc.nguyen,
dinh.phung}@canberra.edu.au

Abstract. User authentication plays an important role in security systems. In general, there are three types of authentications: password based, token based, and biometrics based. Each of them has its own merits and drawbacks. Recently, the research communities successfully explore the possibility that electroencephalography (EEG) being as a new type of biometrics in person recognition, and hence the prospect of using EEG in user authentication is promising. An EEG-based user authentication system has the combined advantages of both password based and biometric based authentication systems, yet without their drawbacks. In this paper we propose to use EEG to authenticate users in multilevel security systems where users are asked to provide EEG signal for authentication by performing motor imagery tasks. These tasks can be single or combined, depending on the level of security required. The analysis and processing of EEG signals of motor imagery will be presented through our experimental results.

Keywords: EEG, data mining, pattern recognition, authentication, security, biometrics.

1 Introduction

Authentication is the foundation of all security systems in which a user is verified if who he or she claims. There are 3 means of authentication: (i) password based also known as something a user knows, for example, password and PIN (personal identity number), (ii) token based as known as something a user has, for example, physical keys, smart cards etc., and (iii) biometrics based as called as something a user is and does such as voice recognition, fingerprints matching, and iris scanning etc. [10].

Authentication by something a user knows is the most popular authentic mechanism, where a user has to provide not only ID but also a password [3]. The system is simple, accurate, and effective. However, password based authentication is not immune from malicious attacks. The popular ones are offline dictionary attack, popular password attack, exploiting user mistakes, and exploiting multiple password use [3]. Therefore, a feasible alternative is extremely desirable.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 513–523, 2013.

Authentication by something a user has is an authentic mechanism that is based on objects a user possesses, such as a bank card, a smart card, and a USB Dongle etc.[3]. This kind of authentication requires users always bringing and providing the physical authentication object when accessing the system. Presenting the foreign object causes inconvenience. In additional, tokens can be physically stolen, be duplicated, as well as be hacked by engineering techniques [3]. Securing the tokens is itself a challenge.

Authentication by something a user is and does, also known as biometrics based authentication, tries to authenticate users based on their biometric characteristics. Although biometrics authentication can avoid some disadvantages of password based and token based authentication, the conventional biometrics modalities have some security disadvantages. Face, fingerprint, and iris information can be faked by photography, voice could be recorded, and hand writing may be mimicked [6]. Moreover, individuals can be damaged their biometric characteristics such as finger or face. These disadvantages require a better biometrics for security systems.

In recent years, researchers start to establish the fact that brain-wave patterns are unique to every individual, and thus EEG signals can be used as biometrics measures [9]. Some modeling methods have been used for EEG-based person recognition. Linear support vector machines were employed in [2]. Neural networks with spectral features were used in [16]. In [9], Gaussian mixture models with maximum a posteriori adaptation were applied for person verification. In [15] the author used Auto Regression (AR) coefficients with Principle Component Analysis (PCA) while [20] deployed Fisher's Linear Discriminant (FLD) to reduce the dimensions of AR and power spectrum density (PSD) feature vectors, and then a *k*-nearest neighbors (kNN) classifiers were applied.

More studies in applying machine learning algorithms to EEG-based person recognition can be found in [6], [11-14]. Multi-sphere Support vector data description (MSSVDD) was used in [11,12] while in [7] the author tried to analyse EEG signals based on an ARMA (Auto-Regressive and Moving Average) model.

Most of the current works only analyzed different single mental tasks, and the impact of combined mental tasks has not been investigated. Moreover, those works focused on person recognition rather than person authentication. Although these 2 areas are related to each other, the focus is different. An authentication system requires accuracy and stability, a minimum risk of being faked or information disclosure, non-intrusive, easy to implement and operate, and having different credentials for different levels of security. In addition, the same person may want to set different levels of "EEG passwords" to different levels of security. Therefore, in this paper, we propose to use EEG for authentication in multilevel security systems where users are asked to provide EEG signal for authentication by performing motor imagery tasks. These tasks can be single or combined, depending on the level of security required.

The rest of the paper is organised as follows. In Section 2, we study using EEG for authentication in multilevel security systems. Section 3 describes EEG features. Section 4 describes SVM modeling technique and hypothesis testing. Experiments and results are presented in Section 5. We conclude the paper with a discussion and our future work in Section 6.

2 Using EEG for Authentication in Multilevel Security Systems

While the conventional types of authentication have their own shortcomings as discussed above, EEG emerges as a potential modality for authentication because of the following advantages, yet without shortcomings of the conventional types:

1. EEG is confidential because it corresponds to a secret mental task which cannot be observed;
2. EEG signals are very difficult to mimic because similar mental tasks are person dependent;
3. It is almost impossible to steal because the brain activity is sensitive to the stress and the mood of the person. An aggressor cannot force the person to reproduce the same signals while the subject is under stress [9]; and
4. EEG signals, by nature, require alive person recording [1].

Therefore, we propose an authentication system using EEG signals as illustrated in Figure 1.

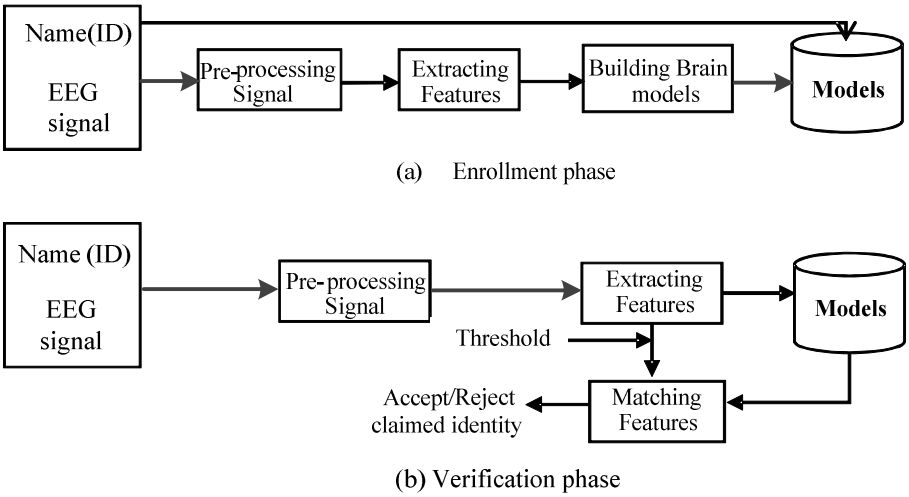


Fig. 1. EEG-based user authentication diagram

We regard the system as authentication by something a user thinks. An EEG based authentication system has two phases: enrolment and verification. In the enrolment phase, a user is asked to do some tasks, for example imagining moving a hand, a foot, a finger or the tongue, and EEG signals are recorded. For authentication purposes, the imagery tasks themselves are also a part of the credentials and could not be seen by any third party. The number of tasks can be flexible and depends on the security level of the system. After collecting the data, the EEG signals of each task corresponding to the user are pre-processed, extracted features, and then the features are used to train the model for this person, which is kept securely in a database.

In the verification phase, when a user wants to access the system, he or she has to provide EEG signals by repeating the tasks which he/she did in the enrolment phase. These input EEG data are processed in the same way as in the enrolment phase. The obtained features are then fed into the classifier as testing data to match the model of the individual who he or she claims to be.

The security systems can have a multiple security levels, depending on zones and resources with EEG based authentication because it can be adjusted by the number of matched tasks. If a system is of a lower security level, an individual may perform a few tasks, and the system only requires that at least one task is matched. If a system is of a high security level, all tasks in the sequence must be matched, so it helps to strength the security system.

In the next sections, we analyse EEG features, choose modelling, and run experiments to illustrate the flexibility of using EEG based authentication in multilevel security systems.

3 EEG Features

3.1 Autoregressive (AR) Features

Autoregressive model can be used for a single-channel EEG signal. It is a simple linear prediction formula that best describes the signal generation system. Each sample $s(n)$ in an AR model is considered to be linearly related, with respect to a number of its previous samples [19]:

$$s(n) = -\sum_{k=1}^{p} a_k s(n-l) + x(n) \tag{1}$$

where a_k, $k = 1, 2, \ldots, p$ are the linear parameters, n denotes the discrete sample time, and $x(n)$ is the noise input. The linear parameters of different EEG channel were taken as the features.

3.2 Power Spectral Density (PSD) Features

Power spectral density (PSD) of a signal is a positive real function of a frequency variable associated with a stationary stochastic process. The PSD is defined as the discrete time Fourier transform (DTFT) of the covariance sequence

$$\emptyset(\omega) = \sum_{k=-\infty}^{\infty} r(k) e^{-i\omega k} \tag{2}$$

where the auto covariance sequence $r(k)$ is defined as

$$r(k) = E\{s(t) y^*(t-k)\} \tag{3}$$

and $s(t)$ is the discrete time signal $\{s(t); t = 0, \pm 1, \pm 2, \ldots\}$ assumed to be a sequence of random variables with zero mean.

In this paper, the Welch's method [21] using periodogram is used for estimating the power of a signal at different frequencies. Twelve frequency components in the band 8-30 Hz of different channels was estimated as features. Welch's method can reduce noise but also reduce the frequency resolution as compared to the standard Bartlett's method, which is desirable for this experiment.

4 Modelling Technique

4.1 Support Vector Machine (SVM)

The training data set obtained during the enrollment phase, is labeled as $\{x_i, y_i\}, i = 1,...,l$, $y_i \in \{-1,1\}$, $x_i \in R^d$. Support vector machine (SVM) using C-Support Vector Classification (C-SVC) algorithm will find the optimal hyperplane $f(x)$ [4]:

$$f(x) = w^T \Phi(x) + b \tag{4}$$

to separate the training data by solving the following optimization problem:

$$min \ \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \tag{5}$$

subject to

$$y_i\left[w^T\Phi(x_i) + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0\ , i = 1,...,l \tag{6}$$

The optimization problem (5) will guarantee to maximize the hyperplane margin while minimizes the cost of error. $\xi_i, i = 1,...,l$ are non-negative, and are being introduced to relax the constraints of separable data problem to the constraint (6) of non-separable data problem. For an error to occur the corresponding ξ_i must exceed unity, so $\sum_i \xi_i$ is an upper bound on the number of training errors. Hence an extra cost $C\sum_i \xi_i$ for errors is added to the objective function where C is a parameter chosen by the user.

In test phase an SVM is used by computing the sign of

$$f(x) = \sum_{i}^{N_S} \alpha_i y_i \Phi(s_i)^T \Phi(x) + b = \sum_{i}^{N_S} \alpha_i y_i K(s_i, x) + b \tag{7}$$

where the S_i are the support vectors, N_S is the number of support vectors, K is kernel with $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$, Φ is a mapping to map the data to some other (possibly infinite dimensional) Euclidean space. One example is Radial Basis Function (RBF) kernel $K(x_i, x_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$.

4.2 Hypothesis Testing

The verification task can be stated as a hypothesis testing between the two hypotheses: the input is from the hypothesis person (H_0), or not from the hypothesis person (H_1).

Let λ_0 is the model of the claimed person and λ_1 is a model representing all other possible people, i.e. impostors. For a given input x and a claimed identity, the choice is between the hypothesis H_0: x is from the claimed person λ_0, and the alternative hypothesis H_1: x is from the impostor λ_1. A claimed person's score $L(x)$ is computed to reject or accept the person claim satisfying the following rules

$$L(x) = \begin{cases} \geq \theta_L & \text{accept} \\ < \theta_L & \text{reject} \end{cases} \tag{8}$$

where θ_L is the decision threshold.

Let x be an EEG feature vector, the probability of x belonging to the class y is defined as $P(x|\theta_y) = ce^{yf(x)}$ where c is normalization factor and $f(x)$ is from (7).

If $x_1, .., x_k$ is a sequence of independent identical density feature vectors of class y, the probability of $x_1, .., x_k$ belonging to the class y in the AND case is:

$$P(x_1, ..., x_k \mid \theta_y) = \prod_{i=1}^{k} c\, e^{yf(x_i)} = c' e^{y\sum_{i=1}^{k} f(x_i)} \tag{9}$$

and its probability belonging to the class y in the OR case is

$$P(x_1, ..., x_k \mid \theta_y) = \max_i c\, e^{yf(x_i)} = ce^{\max_i yf(x_i)} \tag{10}$$

Then the score $L(x)$ in (8) for SVM will become

$$L_{AND}(x) = P(x_1, ..., x_k \mid \theta_1) = c' e^{\sum_{i=1}^{k} f(x_i)} \tag{11}$$

$$L'_{AND}(x) = \sum_{i=1}^{k} f(x_i) \tag{12}$$

$$L_{OR}(x) = P(x_1, ..., x_k \mid \theta_1) = ce^{\max_i yf(x_i)} \tag{13}$$

$$L'_{AND}(x) = \max_i yf(x_i) \tag{14}$$

5 Experiments and Results

5.1 Data Set

The Graz datasets 2008 A and 2008 B come from the Department of Medical Informatics, Institute of Biomedical Engineering, Graz University of Technology, for motor imagery classification problem in BCI Competition 2008 [7]. Both of datasets consist of EEG data from 9 subjects. The subjects were right-handed, had normal or corrected-to-normal vision. In dataset Graz 2008B, the subjects participated

in 5 sessions which consisted of two classes: the motor imagery (MI) of left hand and right hand. Three bipolar recordings (C3, Cz, and C4) were recorded at sampling frequency of 250 Hz. In dataset Graz 2008A, there are four classes motor imagery (MI) of left hand, right hand, foot, and tongue. The subjects participated in two sessions on different days. Each session included 6 runs which were separated by short breaks. Five electrodes (C3, Cz, C4, Fz, and Pz) were recorded at sampling frequency of 250 Hz.

Table 1. Dataset description

Dataset	#subjects	#tasks	#trials	#sessions	Length(secs)
Graz 2008 B	9	2	120	5	7.5
Graz 2008 A	9	4	48	2	6.0

5.2 Feature Extraction

The signals from electrodes C3, C4, Cz were selected for feature extraction. The autoregressive (AR) linear parameters and power spectral density (PSD) components from these signals are extracted as features. In details, the power spectral density (PSD) in the band 8-30 Hz was estimated. The Welch's averaged modified periodogram method was used for spectral estimation. Hamming window was 1 second 50% overlap. There were 12 power components in the frequency band 8-30 Hz extracted.

In addition to PSD features, autoregressive (AR) model parameters were also extracted. In AR model, each sample is considered linearly related with a number of its previous samples. The AR model has the advantage of low complexity and has been used for person identification and authentication [17]. Burg's lattice-based method was used with the AR model order 21, as a previous study [15] suggested when there were many subjects and epochs. The resulting feature set consists of 3*(12+21) = 99 features.

5.3 Results

The SVM method was used to train person EEG models. Experiments were conducted using 5-fold cross validation training and the best parameters found were used to train models on the whole training set and test on a separate test set. The RBF kernel function $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$ was used. The parameters for SVM training are γ and c. The parameter γ was searched in $\{2k: k = -4, -3, \dots, 1\}$. The parameter c was searched in $\{2k: k = -1, -2, \dots, 3\}$. The best parameters found are $c = 8, \gamma = 0.5$.

Due to the levels of security, the task sequence matched is a combination of tasks T_1, T_2, T_3, and T_4 and cases $AND(\wedge)$ and OR $(\vee)$. For example $(T_1 \vee T_2 \vee T_3 \vee T_4)$, or all of them in the right order, e.g. $(T_1 \wedge T_2 \wedge T_3 \wedge T_4)$.

Table 2, Figure 2, Table 3 and Figure 3 present the authentication results when users doing different single motor imagery tasks as well as combined tasks.

Table 2. Equal Error Rate (EER) in authentication of 9 persons B01-B09 in Graz 2008B dataset using the Left hand and Right hand motor imagery tasks

Task / Subject	Left	Right	Left ∪ Right	Left ∩ Right
B01	0.003	0.000	0.001	0.000
B02	0.014	0.002	0.003	0.001
B03	0.000	0.000	0.000	0.000
B04	0.010	0.013	0.001	0.002
B05	0.011	0.010	0.006	0.009
B06	0.005	0.006	0.000	0.003
B07	0.000	0.000	0.000	0.000
B08	0.020	0.013	0.010	0.008
B09	0.000	0.000	0.000	0.000
Average	0.007	0.005	0.002	0.003

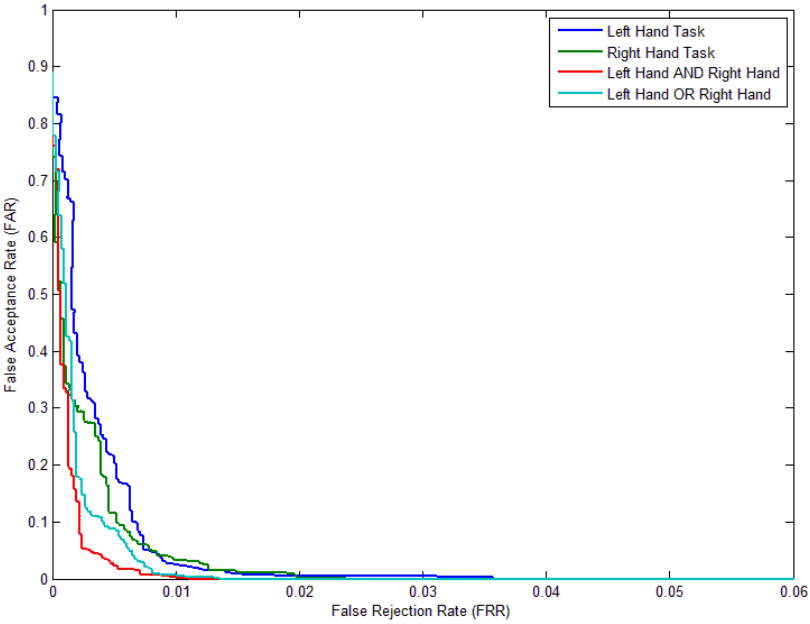


Fig. 2. DET curves of user authentication using EEG signal of Left hand (Task 1), Right hand (Task 2) motor imagery task in Graz 2008B dataset

Table 3. Equal Error Rate (EER) in authentication of 9 persons A01-A09 in Graz 2008A dataset using the Foot and Tongue motor imagery tasks

Task / Subject	Foot	Tongue	Foot ∪ Tongue	Foot ∩ Tongue
A01	0.025	0.000	0.000	0.000
A02	0.033	0.001	0.026	0.012
A03	0.010	0.010	0.000	0.012
A04	0.003	0.000	0.000	0.000
A05	0.003	0.007	0.000	0.010
A06	0.000	0.003	0.000	0.000
A07	0.001	0.001	0.000	0.000
A08	0.000	0.000	0.000	0.000
A09	0.000	0.003	0.000	0.000
Average	0.007	0.004	0.003	0.004

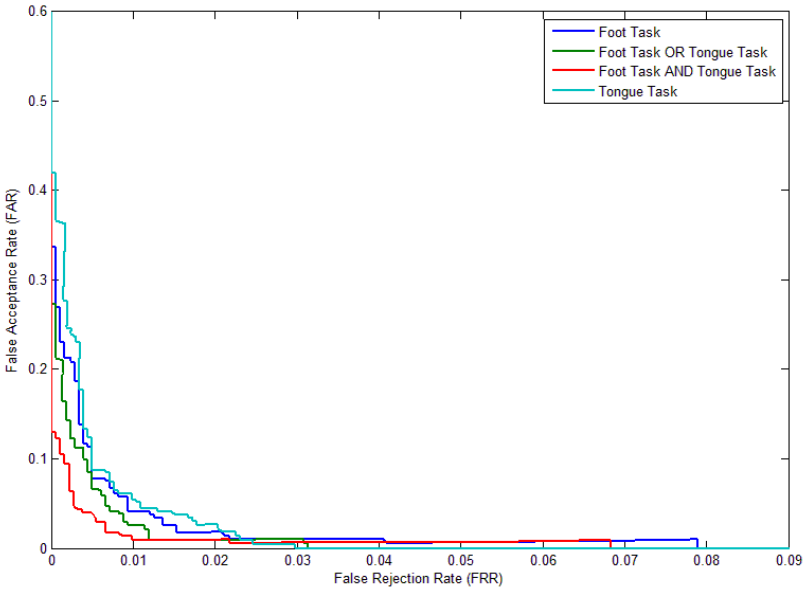


Fig. 3. DET curves of user authentication using EEG signal of Foot (Task 1), Tongue (Task 2) motor imagery task in Graz 2008A dataset

A DET curve is considered as a means of representing performance on detection tasks that involve a trade-off of error types. Therefore, we can see different single mental tasks have different authentication accuracy from Fig 3 and Fig 4. Moreover, the results in Table 2, Table 3 and above DET curves of both datasets confirm that errors are significantly reduced when tasks are combined together in single matched policy ($OR(\vee)$ tasks combination) and multiple matched policy ($AND(\wedge)$ tasks

combination). Moreover, with multiple matched policy ($AND(\wedge)$ tasks combination), it is more difficult to access system that means the security is considerably strengthened. To sum up, EEG-based authentication is flexible and suitable for multilevel security systems.

6 Discussion and Future Work

Using EEG signals for authentication has the advantages of both password based and biometric based authentications, yet without their drawbacks. EEG signals are biometric information of individuals and have the advantages of biometric based authentication, yet EEG based authentication can overcome the disadvantages of conventional biometrics based authentication.

On the other hand, brain patterns correspond to particular mental tasks, and they be regarded as individualized passwords. As the result, EEG based authentication has all the benefits of password based authentication, yet without it's vulnerabilities. Moreover, EEG based authentication provides multilevel security systems and users a flexible authentication mechanism with different single mental tasks as well as different combined tasks policies.

In the near future, we will experiment EEG based authentication on a large dataset. The using EEG signals for remote authentication and potential vulnerabilities of the system will also be studied.

References

1. Allison, B.: Trends in BCI research: progress today, backlash tomorrow? The ACM Magazine for Students 18, 18–22 (2011)
2. Ashby, C., Bhatia, A., Tenore, F., Vogelstein, J.: Low-cost electroencephalogram (EEG) based authentication. In: 2011 5th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 442–445 (2011)
3. Brown, L.: Computer Security: Principles and Practice. William Stallings (2008)
4. Burges, J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
5. Flexer, A.: Data mining and electroencephalography. Statistical Method in Medical Research 9, 395–413 (2000)
6. He, C., Chen, H., Wang, Z.: Hashing the Coefficients from EEG Data for Person Authentication. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, pp. 1445–1448 (2009)
7. Hu, J.: Biometric System based on EEG Signals by feature combination. In: 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 752–755 (2010)
8. Leeb, R., Brunner, C., Muller-Putz, G., Schlogl, A., Pfurtscheller, G.: BCI Competition 2008 - Graz data set B (2008), `http://www.bbci.de/competition/iv/`
9. Marcel, S., Millán, J.R.: Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 743–752 (2007)

10. Ma, W., Campell, J., Tran, D., Kleeman, D.: Password Entropy and Password Quality. In: 2010 4th International Conference on Network and System Security (NSS), 583–587 (2010)
11. Nguyen, P., Tran, D., Le, T., Hoang, T.: Multi-sphere support vector data description for brain-computer interface. In: 2012 Fourth International Conference on Communications and Electronics (ICCE), pp. 318–321 (2012)
12. Nguyen, P., Tran, D., Le, T., Huang, X., Ma, W.: EEG-Based Person Verification Using Multi-Sphere SVDD and UBM. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS, vol. 7818, pp. 289–300. Springer, Heidelberg (2013)
13. Nguyen, P., Tran, D., Huang, X., Sharma, D.: A Proposed Feature Extraction Method for EEG-based Person Identification. In: The International Conference on Artificial Intelligence (ICAI 2012), USA (2012)
14. Nguyen, P., Tran, D., Huang, X., Ma, W.: Motor Imagery EEG-Based Person Verification. In: Rojas, I., Joya, G., Cabestany, J. (eds.) IWANN 2013, Part II. LNCS, vol. 7903, pp. 430–438. Springer, Heidelberg (2013)
15. Palaniappan, R.: Two-stage biometric authentication method using thought activity brain waves. International Journal of Neural Systems 18 (2008)
16. Poulos, M., Rangoussi, M., Alexandris, N.: Neural network based person identification using EEG features. In: Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1999, pp. 1117–1120 (1999)
17. Poulos, M., Rangoussi, M., Alexandris, N., Evangelou, A.: Person identification from the EEG using nonlinear signal classification. Methods of Information in Medicine 41, 64–75 (2002)
18. Riera, A., Soria-Frisch, A., Caparrini, M., Grau, C., Ruffini, G.: Unobtrusive biometric system based on electroencephalogram analysis. EURASIP Journal on Advances in Signal Processing 2008 (2008)
19. Sanei, S., Chambers, J.: EEG signal processing. Wiley-Interscience (2007)
20. Yazdani, A., Roodaki, A., Rezatofighi, S.H., Misaghian, K., Setarehdan, S.K.: Fisher linear discriminant based person identification using visual evoked potentials. In: 9th International Conference on Signal Processing, ICSP 2008, pp. 1677–1680 (2008)
21. Welch, P.: The use of Fast Fourier Transform for the estimation of power spectra: a method based on time averaging over short, modified periodogram. IEEE Trans. Audio Electroacoustics, 70–73 (1967)

A New Fuzzy Extreme Learning Machine for Regression Problems with Outliers or Noises

Enhui Zheng*, Jinyong Liu, Huijuan Lu, Ling Wang, and Le Chen

China Jiliang University, College of Mechanical and Electrical Engineering, Hangzhou 310018, Zhejiang province, P.R. China
ehzheng@cjlu.edu.cn, hzliujy@126.com

Abstract. Extreme Learning Machine (ELM), recently proposed by Huang et al., has attracted much attention from more and more researchers in the machine learning and data mining community, and has shown similar or better generalization performance with dramatically reduced training time than Support Vector Machines (SVM). In ELM, it is implicitly assumed that all samples in training datasets share the same importance. Therefore, when it comes to datasets with outliers or noises, like SVM, ELM may produce suboptimal regression models due to overfitting. In this paper, by equipping ELM with the fuzzy concept, we propose a novelty approach called New Fuzzy ELM (NF-ELM) to deal with the above problem. In NF-ELM, firstly, different training samples are assigned with different fuzzy-membership values based on their degree of being outliers or noises. Secondly, these membership values are incorporated into the ELM algorithm to make it less sensitive to outliers or noises. The performance of the proposed NF-ELM algorithm is evaluated on three artificial datasets and thirteen real-world benchmark function approximation problems. The results indicate that the proposed NF-ELM algorithm achieves better predictive accuracy in most cases than ELM and SVM does.

Keywords: Extreme learning machine, Fuzzy membership, Regression, Outliers or noises.

1 Introduction

Extreme learning machine (ELM), recently developed by Huang et al. [1, 2], is a non-iterative learning algorithm for single-hidden-layer feed-forward networks (SLFNs). In ELM, the parameters (input weights and hidden bias) of hidden nodes are randomly generated and, the output weights connecting the hidden to the output are analytically determined by computing a simple least-squares solution rather than a time-consuming optimization [1-5]. ELM is originally developed for the SLFNs and then extended to the "generalized" SLFNs which may not be neuron alike [3, 4]. The related work of ELM is overviewed in [5].

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 524–534, 2013.

Because of its similar or better generalization performance and much faster learning speed than conventional learning machines such as Support Vector Machines (SVM), ELM has been attracting more and more attentions in the machine learning and data mining community. Although many improved versions [6-17] of ELM have been presented, such as Prune ELM [6], Optimal Prune ELM [12, 13], Regularized ELM [11], Evolutionary ELM [14], Kernel based ELM [7], Two-stage ELM [8], Ensemble based ELM [15], Incremental ELM [10], Error Minimized ELM [9], Evolutionary Fuzzy ELM [16], Online Sequential Fuzzy ELM [17], and so on, to enhance the performance of ELM, ELM and its improved versions still suffer from the problems of outliers or noises due to overfitting.

It should be noted that all samples in training datasets are regarded as the same importance in ELM and its improved versions mentioned above. However, in many real regression applications, some samples in the training datasets may inevitably be corrupted in various degrees by outliers or noises. In general, like most of the regression algorithms, ELM establishes the predictive model by minimizing the global error in the training phase, which means that the model needs to fit all samples in training datasets well, including the outliers or noises. Therefore, the overfitting occurs, and the generalization performance of the trained model is degraded.

In this paper, inspired by the ideas of Fuzzy SVM [18-24], we propose a novelty approach termed as New Fuzzy ELM (NF-ELM), in which ELM is equipped with fuzzy concepts to cope with the problems of outliers or noises. In NF-ELM, according to the degree of samples being contaminated, we first assign different training samples with different fuzzy-membership values, where more important samples are assigned with higher membership values while less important ones are assigned with lower membership values. Then NF-ELM is presented by integrating those fuzzy-membership values into ELM to reduce the effect of outliers or noises in the learning of a regression model.

We evaluate the proposed NF-ELM method on three artificial and thirteen real-world benchmark datasets and, compare its performance with the ELM, SVM and BP algorithms. It is demonstrated that the proposed NF-ELM algorithm obtains better generalization performance for regression problems in the presence of outliers or noises.

This paper is organized as follows. Section 2 briefly reviews the ELM algorithm. A new algorithm, called NF-ELM is proposed and deduced in section 3. In section 4, the experiments are conducted and the results are compared with other methods. Finally, section 5 concludes this paper.

2 Fuzzy ELM

In this section, we propose a new approach, termed as NF-ELM, in which ELM is equipped with fuzzy concepts to cope with the regression problems on datasets present of outliers or noises. The formulation and deduction are presented in detail as follows.

2.1 Formulation of NF-ELM

Suppose that we have N arbitrary distinct samples

$$\{(x_j, y_j, w_j)\}_{j=1}^{N},\ x_j \in R^d,\ y_j \in R^m,\ w_j \in R, \tag{7}$$

where w_j, are the fuzzy-membership value of sample x_j toward the regression model, $j = 1, \cdots, N$. In the following, we reformulate a new method of training a SLFN, called NF-ELM, based on the given number of hidden nodes L and an activation function $g(x)$.

$$\begin{bmatrix} e_1 \\ \vdots \\ e_j \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} o_1 - y_1 \\ \vdots \\ o_j - y_j \\ \vdots \\ o_N - y_N \end{bmatrix} = \begin{bmatrix} h(x_1) \cdot \beta - y_1 \\ \vdots \\ h(x_j) \cdot \beta - y_j \\ \vdots \\ h(x_N) \cdot \beta - y_N \end{bmatrix} \tag{8}$$

To show the effect of fuzzy-membership values on the learning of regression model clearly, the system (2) of ELM is rewritten as (7). It can be seen, from (8), that each error e_j, $j = 1, \cdots, N$, is treated as equally important in ELM.

However, the importance of different samples may be different in real-world applications. For example, the samples contaminated by outliers and noises could be made less important to avoid overfitting. For $j = 1, \cdots, N$, the degree of importance of the sample x_j can be defined by its fuzzy-membership value w_j that is the attitude of x_j toward the regression model.

By multiplying the error e_j of the sample x_j by the corresponding fuzzy-membership value w_j, $j = 1, \cdots, N$, the system (9) can be constructed as

$$\begin{bmatrix} w_1 e_1 \\ \vdots \\ w_j e_j \\ \vdots \\ w_N e_N \end{bmatrix} = \begin{bmatrix} w_1 (o_1 - y_1) \\ \vdots \\ w_j (o_j - y_j) \\ \vdots \\ w_N (o_N - y_N) \end{bmatrix} = \begin{bmatrix} w_1 (h(x_1) \cdot \beta - y_1) \\ \vdots \\ w_j (h(x_j) \cdot \beta - y_j) \\ \vdots \\ w_N (h(x_N) \cdot \beta - y_N) \end{bmatrix}. \tag{9}$$

In this formulation, as done in FSVM [18-24], the fuzzy-membership value w_j of the sample x_j is integrated into the system (9). If the sample x_j is a noisy sample that should be treated as the less important, we assign it with a smaller w_j to reduce the effect of the error e_j on the learning of the regression model, and therefore, the contribution of the corresponding x_j to the trained regression model is decreased.

In another view, the proposed NF-ELM could achieve the better generation performance by letting the larger training error of less important examples, such as outliers or noise.

Those equations in (10) can be compactly rewritten as

$$WE = W(H\beta - Y) = WH\beta - WY, \tag{10}$$

where $W=\mathrm{diag}(w_1, w_2, \cdots, w_N)$,

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(a_1,b_1,x_1) & \cdots & G(a_L,b_L,x_1) \\ \vdots & \vdots & \vdots \\ G(a_1,b_1,x_N) & \cdots & G(a_L,b_L,x_N) \end{bmatrix}_{N\times L} \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times \mathrm{m}} \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times m} \tag{11}$$

Further, the equations in (11) can be transformed into

$$WE = H_F\beta - Y_F, \tag{12}$$

where $H_F = WH$ and $Y_F = WY$.

It can be seen that both the characteristic and the frame of the presented NF-ELM stay the same with ELM, except for the fuzzy-membership added. Therefore, the output weights of NF-ELM for SLFNs can be analytically determined, by the least-squares method, as

$$\hat{\beta}_F = \arg\min_{\beta} \left\| H_F\beta - Y_F \right\| = H_F^{\dagger} Y_F, \tag{13}$$

where $H_F^{\dagger}$ is called the Moore-Penrose generalized inverse of matrix H_F. Moreover, if $H_F^T H_F$ is non-singular, $H_F^{\dagger} = (H_F^T H_F)^{-1} H_F^T$, while if $H_F H_F^T$ is non-singular, $H_F^{\dagger} = H_F^T (H_F H_F^T)^{-1}$.

According to the ridge regression theory [1, 5], one can add a positive value λ to the diagonal of $H_F^T H_F$ or $H_F H_F^T$ to make the resultant solution more stable, and thus, the generalization performance of trained regression model can be enhanced. Therefore, If $H_F^T H_F$ is non-singular, the regression model is

$$f(x) = h(x) H_F^T (I/\lambda + H_F H_F^T)^{-1} Y_F, \tag{14}$$

while if $H_F H_F^T$ is non-singular, the regression model is

$$f(x) = h(x) (I/\lambda + H_F^T H_F)^{-1} H_F^T Y_F. \tag{15}$$

Thus, The NF-ELM algorithm proposed in this paper can be summarized as follows.

NF-ELM Algorithm: Given training set $\{(x_j, y_j, w_j)\}_{j=1}^N$, where $x_j \in R^d$, $y_j \in R^m$ and $w_j \in R$, the hidden node output function $g(x)$, and the hidden node number L.

Step1: Randomly generate hidden node parameters $\{(a_i, b_i)\}_{i=1}^L$.

Step2: Calculate the hidden layer output matrix H.

Step3: Construct the matrix $W=\mathrm{diag}(w_1, w_2, \cdots, w_N)$.

Step4: Calculate $H_F = WH$ and $Y_F = WY$.

Step5: Calculate the output weights

$$\hat{\beta} = H_F^T(\ H_F H_F^T)^{-1} Y_F \quad \text{or} \quad \hat{\beta} = (\ H_F^T H_F)^{-1} H_F^T Y_F .$$

3 Generating Fuzzy-Membership Values

It is an important step to generate fuzzy-membership values for each training sample before analytically determining the output weight according to (13) by the least-squares method. Before training the proposed NF-ELM model, we must appropriately choose or define the membership generation method to reflect the importance of each training sample. The proper definition or application of membership function is depended on the characteristic of the certain field or task. The undesirable membership settings will send wrong messages to the training process and degrade the generalization performance of trained regression model.

In order to define the fuzzy-membership function, which gives the particular importance of each example, we consider three methods in the following section.

3.1 Defining Fuzzy-Membership Function Based on Time Characteristic

Consider that the sequential learning problem needs to be conducted in some real-world applications, where the arriving time of the sample x plays an decisive role in determining its importance, one can define the fuzzy-membership value w_j of the sample x_j with respect to its arriving time t_j. In general, the sample from recent past is more important than the one far back in the past in some applications involving real-time signal processing [18]. Therefore, when training the regression model, the sample x_N that is the last to reach the system will be made the most important, and its fuzzy-membership values is determined as $w_N = 1$, while the first arriving sample x_1 will be made the least important, and its fuzzy-membership values is chosen as $w_1 = \sigma$, in which the positive value σ is the user-defined lower bond of fuzzy-membership values. Further, suppose the fuzzy-membership values are a linear function of time, as defined in [18], the fuzzy-membership function can be defined as

$$w_j = f_1(t_j) = \frac{1-\sigma}{t_N - t_1} t_j + \frac{t_N \sigma - t_1}{t_N - t_1}, \tag{16}$$

where $t_1 < \cdots t_j < \cdots t_N$ is the sequence of training samples obtained in the system.

3.2 Defining Fuzzy-Membership Function Based on the Distribution Density

In this method, we assume that the fuzzy-membership values could depend mainly on the space distribution of training samples. Further, we consider that the samples closer to the position around which the density of samples is low are treated as outliers or

noises with more possibility. The lower density position around which the sample lies, the less important the sample is regarded as. Therefore, the fuzzy-membership function values w_j of sample x_j can be defined with respect to the average Euclidean distance d_j of x_j to its k nearest neighbors, $j = 1, \cdots, N$, where the parameter k is user-specified in advance. In the cases that the fuzzy-membership value w_j is a linear function of the distance d_j, the fuzzy-membership function can be constructed in this paper as

$$w_j = f_2(d_j) = 1 - \frac{d_j}{d_{ma} + \Delta}, \tag{17}$$

where $d_{ma} = \max\{d_1, \cdots, d_j, \cdots, d_N\}$, and the parameter Δ is a small positive value user-defined in advance.

3.3 Defining Fuzzy-Membership Function Based on the Training Error

In this method, to avoid overfitting, we consider that the training error of each sample could play a decisive role in determining its importance, and that the samples with larger errors are treated as outliers or noises with more possibility. For each training sample, the smaller the training error is, the more important the sample is regarded as, while the larger the training error is, the less important the sample is regarded as. Therefore, we can define the fuzzy-membership function values w_j of sample x_j by a linear function

$$w_j = f_3(e_j) = 1 - \frac{e_j}{e_{ma} + \Delta}, \tag{18}$$

where $e_{ma} = \max\{e_1, \cdots, e_j, \cdots, e_N\}$, Δ is a small positive value user-defined in advance, and the parameter e_j can be calculated based on a trained normal ELM model according to

$$e_j = h(x_j) \cdot \hat{\beta} - y_j. \tag{19}$$

4 Experimental Results

To evaluate the performance of the presented NF-ELM algorithm for regression problems in presence of outliers or noises, the experiments on three artificial and thirteen real-world benchmark datasets are conducted in following sections. All of the simulations are carried out in a Matlab environment on an intel core 2 Duo, 2.53 GHZ PC with 2 GB memory. In all experiments, we set $\Delta = 0.001$ and $\lambda = 0.0001$.

The index used for evaluating the performance is the Root Mean Square Error (RMSE). In this paper, the NF-ELM is implemented by embedding the fuzzy-membership values into the ELM software that can be downloaded from http://www.ntu.edu.sg/home/egbhuang/.

4.1 Experiments on Three Artificial Datasets

In this section, we adopt three examples to evaluate the performance of the proposed NF-ELM in terms of the predictive error of each sample.

In the first example, we construct a symmetrical function defined as

$$y = \begin{cases} -6x, & -10.05 \le x \le -0.15, \\ 0, & -0.15 < x < 0.15, \\ 6x, & 0.15 \le x \le 10.05, \end{cases} \tag{21}$$

In this example, thirty two training samples are uniformly generated with $x \in [-10.05, -0.15] \cup [0.15, 10.05]$. By the same way, the sixty two test samples are generated. Consider that the samples from recent past is more important than the samples far back in the past in some applications, fuzzy-membership is defined as

$$w_t = \begin{cases} 0.5, & -10.05 \le x \le -0.15, \\ 1, & 0.15 \le x \le 10.05. \end{cases} \tag{22}$$

It can be noted that the definition in (22) is the reduced version of (16). It can be easily checked that the errors in ELM are almost symmetrical with respect to $x = 0$, while in NF-ELM, average value of errors when $0.15 \le x \le 10.05$ are much smaller than that of errors when $-10.05 \le x \le -0.15$. Therefore, we can achieve the goal of making the samples from recent past be more important than the samples far back in the past when establishing a predictive model, which is usually necessary in some domains such as the time series prediction.

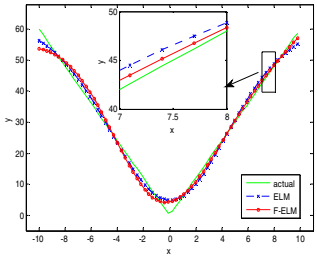


Fig. 1. Test results of ELM and F-ELM in function (21)

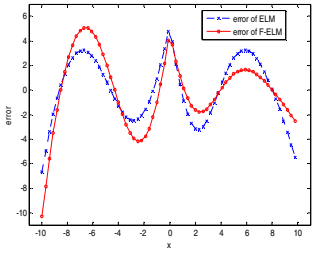


Fig. 2. Test errors of ELM and F-ELM in function (21)

In the second example, we consider a simple function defined as

$$y = \begin{cases} \sin x / x & x \ne 0 \\ 1 & x = 0 \end{cases}. \tag{23}$$

In this example, sixty eight training samples $\{x_i, y_i\}_{i=1}^{68}$ are uniformly generated with $x \in [-10, 10]$. Before training the decision model, for $i = 30, 32, 35$, we

execute $y_i = y_i + 0.3$ to construct three outliers or noises , as shown in Figure 3. Ninety six test samples are uniformly extracted from (23) with the same interval.

Different from the training samples, No outliers or noises exist in test samples. In NF-ELM and ELM, the number of hidden nodes is chosen as 5. The results are described in Figure 3, Figure 4, respectively. It is demonstrated that the NF-ELM yields more accurate predictive performance than ELM does when the training samples are corrupted by outliers or noises.

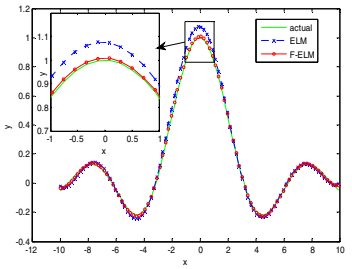


Fig. 3. Test results of ELM and F-ELM in function (23)

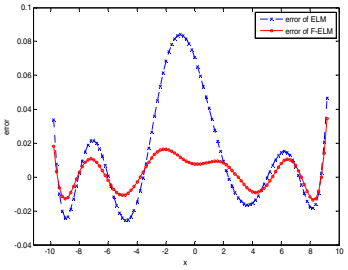


Fig. 4. Test results of ELM and F-ELM in function (23)

In the third example, we consider a two-variable function as

$$y = \frac{(5-x_2)^2}{3(5-x_1)^2 + (5-x_2)^2} + \text{Noise}\,,\ 0 \le x_1, x_2 \le 10\,. \tag{24}$$

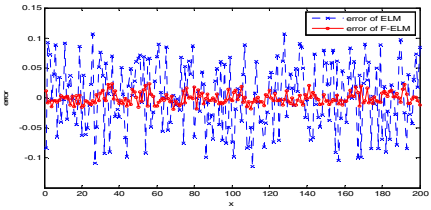


Fig. 5. Test errors of ELM and F-ELM in function (24)

In this example, a training dataset containing 100 samples is uniformly generated from the input domain with a Gaussian noise [0, 0.25], and, by the same way, we obtain the test dataset consisting of 200 samples, in which no outliers or noises exist. In ELM and NF-ELM, the number of hidden nodes L is set as 6. The results are depicted in Figure 5. It is illustrated that the NF-ELM achieves much smaller test errors than ELM does.

4.2 Experiments on Real Benchmark Datasets

In this section, we consider thirteen real-world benchmark datasets used in literature [1] to evaluate the proposed NF-ELM method. The performance of NF-ELM is compared with ELM. It should be noted that the computational time of the proposed NF-ELM includes the process of calculating the fuzzy membership values of training samples and the process of learning the decision model based on least square method. Therefore, the computational time of the NF-ELM is little longer than that of the normal ELM algorithm. Note that, proportion of noisy samples in training dataset is determined by specific problems. Therefore, when the problem is determined, the proportion of noisy samples in training dataset is determined too. In order to explain the conclusion above, a illustrate about the connection between proportion of noisy samples and test accuracy is given.

Table 1. Training and test RMSE of ELM, FELM2 and FELM3

Data sets	ELM [1]		0.90NF-ELM2		0.90NF-ELM3		0.95NF-ELM2		0.95NF-ELM3	
	train	test	train	test	train	test	train	test	train	test
Auto price	0.075	0.099	0.084	0.099	0.075	0.097	0.075	0.096	0.075	**0.096**
Servo	0.071	0.120	0.070	**0.115**	0.069	0.118	0.072	0.118	0.067	0.116
Triazines	0.190	0.220	0.190	0.213	0.135	0.208	0.126	**0.201**	0.229	0.209
Breast cancer	0.247	0.268	0.208	0.267	0.215	0.266	0.218	0.267	0.173	**0.204**
Machine cpu	0.033	0.054	0.031	0.052	0.033	0.055	0.033	**0.041**	0.032	0.046
Stocks	0.025	0.035	0.034	0.035	0.022	**0.033**	0.035	0.034	0.014	0.034
Abalone	0.080	0.082	0.079	0.082	0.070	0.080	0.080	0.082	0.070	**0.080**
ailerons	0.042	0.043	0.045	0.042	0.043	**0.039**	0.042	0.042	0.043	0.039
Computer	0.032	0.038	0.033	0.038	0.039	0.038	0.031	0.038	0.039	**0.036**
elevators	0.055	0.057	0.054	**0.055**	0.055	0.057	0.053	0.057	0.055	0.057
Bank	0.041	0.036	0.042	0.035	0.040	0.036	0.041	**0.034**	0.040	0.036
housing	0.122	0.127	0.121	0.122	0.110	**0.123**	0.113	0.122	0.109	0.124
Census(8L)	0.062	0.066	0.064	0.066	0.064	0.066	0.063	0.066	0.064	**0.065**

5 Conclusion

In this paper, a novel ELM approach, named as NF-ELM, is proposed to overcome the problems of outliers or noises encountered by ELM. In NF-ELM, we assign low fuzzy-membership values for training samples contaminated by outliers or noises, and thus their contributions to the learning of regression model are decreased. The performance of NF-ELM is evaluated on three artificial datasets and thirteen real-world datasets, and is compared with ELM of the RSME. In general, we can conclude that

NF-ELM (NF-ELM2 or NF-ELM3) achieves higher predictive accuracy compared to the above three algorithms when learning a regression model from the datasets in presence of outliers or noises in this paper. However, it should be noted that the fuzzy-membership function adopted in NF-ELM plays an important role in obtaining the model with better generalization. Moreover, the optimal membership function is dataset-dependent.

Acknowledgements. This work is partly supported by the National Natural Science Foundation of China (60905034) and Natural Science Foundation of Zhejiang province (Y1080950 and Y1080918) and Project of Zhejiang Provincial Department of Education (Y201121959)

References

1. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing 70, 489–501 (2006)
2. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of International Joint Conference on Neural Networks, pp. 25–29 (2004)
3. Huang, G.-B., Chen, L.: Convex incremental extreme learning machine. Neurocomputing 70, 3056–3062 (2007)
4. Huang, G.-B., Chen, L.: Enhanced random search based incremental extreme learning machine. Neurocomputing 70, 3460–3468 (2008)
5. Huang, G.-B., Wang, D.-H., Lan, Y.: Extreme learning machine: a survey. Int. J. Mach. Learn. & Cyber. (2), 107–122 (2011)
6. Rong, H.-J., Ong, Y.-S., Tan, A.-H., Zhu, Z.: A fast pruned extreme learning machine for classification. Neurocomputing 72, 359–366 (2008)
7. Huang, G.-B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multi-class classification. IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics (2011)
8. Lan, Y., Soh, Y.-C., Huang, G.-B.: Two-stage extreme learning machine for regression. Neurocomputing 73, 223–233 (2010)
9. Feng, G., Huang, G.-B., Lin, Q., Gay, R.: Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans. Neural. Netw. 20, 1352–1357 (2009)
10. Huang, G.-B., Chen, L., Siew, C.-K.: Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71, 576–583 (2008)
11. Deng, W.-Y., Zheng, Q.-L., Chen, L.: Regularized extreme learning machine. IEEE Symposium on Computational Intelligence and Data Mining (2), 389–395 (2009)
12. Miche, Y., Sorjamaa, A., Lendasse, A.: OP-ELM: theory, experiments and a toolbox. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 145–154. Springer, Heidelberg (2008)
13. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: optimally pruned extreme learning machine. IEEE Trans. Syst. Man Cybern. Part B Cybern. 21, 158–162 (2010)
14. Zhu, Q.-Y., Qin, A.-K., Suganthan, P.-N., Huang, G.-B.: Evolutionary extreme learning machine. Pattern Recognition 38, 1759–1763 (2005)

15. Liu, N., Han, W.: Ensemble based extreme learning machine. IEEE Singal Processing Letters 17, 754–757 (2010)
16. Qu, Y.-P., Shang, C.-J., Wu, W., Shen, Q.: Evolutionary fuzzy extreme learning machine for mammographic risk analysis. International Journal of Fuzzy Syetems 13, 282–291 (2011)
17. Rong, H.-J., Huang, G.-B., Sundararajan, N., Saratchandran, P.: Online sequential fuzzy extreme learning machine for function approximation and classification problems. IEEE Transactions on System, Man and Cybernetics-Part B: Cybernetics 39, 1067–1072 (2009)
18. Lin, C.-F., Wang, S.-D.: Fuzzy support vector machines. IEEE Transactions on Neural Networks 5, 2603–2613 (2002)
19. Huang, H.-P., Liu, Y.-H.: Fuzzy support vector machines for pattern recognition and data mining. Int. J. Fuzzy Sys. 4, 826–835 (2002)
20. Jiang, X.-F., Yi, Z., Lv, J.-C.: Fuzzy SVM with a new fuzzy membership function. Neural Comput. Appl. 15, 268–276 (2006)

Author Index