

Jie Tang
Irwin King
Ling Chen
Jianyong Wang (Eds.)

LNAI 7120

Advanced Data Mining and Applications

7th International Conference, ADMA 2011
Beijing, China, December 2011
Proceedings, Part I

1
Part I

 Springer

Lecture Notes in Artificial Intelligence 7120

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Jie Tang Irwin King Ling Chen
Jianyong Wang (Eds.)

Advanced Data Mining and Applications

7th International Conference, ADMA 2011
Beijing, China, December 17-19, 2011
Proceedings, Part I

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Jie Tang
Jianyong Wang
Tsinghua University
Department of Computer Science and Technology
Beijing, 100084, China
E-mail: {jietang, jianyong}@tsinghua.edu.cn

Irwin King
The Chinese University of Hong Kong
Department of Computer Science and Engineering
Hong Kong, SAR, China
E-mail: king@cse.cuhk.edu.hk

Ling Chen
University of Technology
Faculty of Engineering and Information Technology
Sydney, NSW 2007, Australia
E-mail: ling.chen@uts.edu.au

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-25852-7 e-ISBN 978-3-642-25853-4
DOI 10.1007/978-3-642-25853-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2, H.3, H.4, H.2.8, J.1, F.1, I.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

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

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

The continuous growth of digital technologies leads to not only the availability of massive amounts of data, but also the emergence of new types of data with novel characteristics. It poses new challenges for the data-mining research community to develop sophisticated data-mining algorithms as well as successful data-mining applications. For the purpose of promoting the original research in advanced data mining and applications, bringing together the experts on data mining throughout the world, and providing a leading international forum to exchange research ideas and results in emergent data-mining problems, the 7th International Conference on Advanced Data Mining and Applications was held in Beijing, China, in 2011.

The conference received 191 paper submissions from 47 countries and areas. All papers were peer reviewed by at least three members of the Program Committee (PC) composed of international experts in data-mining fields, as well as one Vice PC Co-chair. The PC, together with our PC Co-chairs, worked very hard to select papers through a rigorous review process and extensive discussion, and finally composed a diverse and exciting program including 35 full papers and 29 short papers. The ADMA 2011 program was highlighted by three keynote speeches from outstanding researchers in advanced data-mining and application areas: Philip S. Yu (University of Illinois Chicago), Wolfgang Nejdl (L3S Research Center), and Stefan Decker (National University of Ireland).

Without the support of several funding agencies and organizations, the successful organization of the ADMA 2011 would not be possible. These include sponsorships from: IBM Research, China Samsung Telecom R&D Center, and Tsinghua University. We would also like to express our gratitude to the General Co-chairs for all their precious advice and the Organizing Committee for their dedicated organizing efforts. Last but not least, we sincerely thank all the authors, presenters and attendees who jointly contributed to the success of ADMA 2011!

December 2011

Jie Tang
Irwin King
Ling Chen
Jianyong Wang

Finance chair

Peng Cui Tsinghua University, China

Publicity Co-chairs

Juanzi Li Tsinghua University, China
Zhichun Wang Tsinghua University, China
Jibing Gong Chinese Academy of Sciences, China

Industrial Track Chair

Keke Cai IBM Research, China

Registration Chair

Xiaonan Liu Tsinghua University, China

Special Track Co-chairs

Hongyan Liu Tsinghua University, China
Zhong Su IBM Research, China

Web Master

Bo Gao Tsinghua University, China

Program Committee

Vice PC Chairs

Ling Chen, Australia	Wei Chen, China
Hong Chen, Hong Kong	Bin Cui, China
Xiaoyong Du, China	Ruoming Jin, USA
Zhan-huai Li, China	Xiaofeng Meng, China
Marie-Francine Moens, Belgium	Kyuseok Shim, Korea
Yizhou Sun, USA	Wei Wang, China
Hao Wang, USA	Ying Zhao, China
Aoying Zhou, China	

PC Members

Aijun An, Canada	Aixin Sun, Singapore
Akihiro Inokuchi, Japan	Alfredo Cuzzocrea, Italy
Ali Daud, China	Amanda Clare, UK
Andrzej Skowron, Poland	Annalisa Appice, Italy
Atsuyoshi Nakamura, Japan	Brijesh Verma, Australia
Bruno Cremilleux, France	Chenhao Tan, USA
Cheqing Jin, China	Chi Wang, USA
Chiranjib Bhattacharyya, India	Chotirat Ratanamahatana, Thailand
Chun-Hung Li, Hong Kong	Chun-Nan Hsu, USA
Cindy Lin, USA	Daisuke Ikeda, Japan

Daisuke Kawahara, Japan
Daoqiang Zhang, China
David Taniar, Australia
Di Wu, China
Dianhui Wang, Australia
Du Zhang, USA
Faizah Shaari, Malaysia
Fusheng Yu, China
Gang Li, Australia
Guohe Li, China
Guoqiong Liao, China
Hanghang Tong, USA
Harry Zhang, Canada
Hongan Wang, China
Hongzhi Wang, China
Huan Huo, China
Huidong Jin, Australia
James Bailey, Australia
Jan Rauch, Czech Republic
Jiakui Zhao, China
Jianhui Chen, USA
Jibing Gong, China
Jinbao Li, China
Jing Liu, China
Jizhou Luo, China
Keke Cai, China
Kritsada Sriphaew, Japan
Lian Yu, China
Licheng Jiao, China
Lisa Hellerstein, USA
Mao Ye, China
Mario Linares-Vásquez, Colombia
Masashi Sugiyama, Japan
Mengjie Zhang, New Zealand
Michael Madden, Ireland
Michele Berlingerio, Italy
Min Yao, China
Ming Li, China
Ming-Syan Chen, Taiwan
Nicolas Spyrtos, France
Odysseas Papapetro, Greece
Panagiotis Karras, Singapore
Philippe Fournier-Viger, Taiwan
Qinbao Song, China
Qingshan Liu, China
Danzhou Liu, USA
Dao-Qing Dai, China
Dexi Liu, China
Diane Cook, USA
Donghui Zhang, USA
Eduardo Hruschka, Brazil
Feiping Nie, USA
Gaël Dias, Portugal
George Karypis, USA
Guojie Song, China
Guoyin Wang, China
Hanzi Wang, China
Hassan Abolhassani, Iran
Hongyan Li, China
Hua Lu, Denmark
Hui Xiong, USA
Hung Son Nguyen, Poland
James Kwok, Hong Kong
Jason Wang, USA
Jian Yin, China
Jian-Min Han, China
Jimmy Huang, Canada
Jing Gao, USA
Jinghai Rao, China
K. Selcuk Candan, USA
Kitsana Waiyamai, Thailand
Li Li, China
Liang Sun, USA
Lidan Shou, China
Manish Gupta, USA
Marco Maggini, Italy
Martine De Cock, Belgium
Masayuki Numao, Japan
Michael Bain, Australia
Michalis Vazirgiannis, Greece
Michele Sebag, France
Ming Ji, China
Mingchun Wang, Taiwan
Nicola Di Mauro, Italy
Ninghui Li, USA
Pablo Castro, Argentina
Patrick Gallinari, France
Qi Wang, China
Qing He, China
Ran Wolff, Israel

Ravi Kumar, USA
 Sadok Ben Yahia, Tunisia
 Sang-Hyuk Lee, Korea
 Sanparith Marukatat, Thailand
 Sheng Zhong, USA
 Shengtao Sun, China
 Shu-Ching Chen, USA
 Shuliang Wang, China
 Songhua Xu, USA
 Stefan Skudlarek, Japan
 Sung Ho Ha, Republic of Korea
 Takehisa Yairi, Japan
 Tao Qin, China,
 Thepchai Supnithi, Thailand
 Tim Weninger, USA
 Tomonari Masada, Japan
 Tru Cao, Vietnam
 Tu-Anh Nguyen-Hoang, Vietnam
 Wai Lam, Hong Kong
 Weizhu Chen, China
 Wenyang Bai, China
 Wlodek Zadrozny, USA
 Xiangliang Zhang, Saudi Arabia
 Xiaohui Liu, UK
 Xin Jin, USA
 Xintao Wu, USA
 Xue Li, Australia
 Yang Xiang, USA
 Yasuhiko Morimoto, Japan
 Yi Chen, USA
 Yihua Wu, USA
 Yi-Ping Chen, Australia
 Yongli Wang, China
 Yubao Liu, China
 Yuhua Li, China
 Zhihai Wang, China
 Zhipeng Xie, China
 Zhongzhi Shi, China
 Zi Yang, China
 Zili Zhang, Australia

Rui Camacho, Portugal
 Sai Wu, Singapore
 Sanjay Jain, Singapore
 Shen Jialie, Singapore
 Shengfei Shi, China
 Shichao Zhang, China
 Shuigeng Zhou, China
 Songcan Chen, China
 Srikanta Tirthapura, USA
 Stefano Ferilli, Italy
 Tadashi Nomoto, Japan
 Tao Li, USA
 Tetsuya Yoshida, Japan
 Tieyun Qian, China
 Tomoharu Iwata, Japan
 Toshiro Minami, Japan
 Tsuyoshi Murata, Japan
 Wagner Meira Jr., Brazil
 Wei Liu, Australia
 Wen-Chih Peng, Taiwan
 Wilfred Ng, Hong Kong
 Wynne Hsu, Singapore
 Xiaohua Hu, USA
 Xide Lin, USA
 Xingquan Zhu, Australia
 Xiuli Ma, China
 Xuelong Li, China
 Yang-Sae Moon, Republic of Korea
 Yasuhito Asano, Japan
 Yifeng Zeng, Denmark
 Ying Zhang, Australia
 Yonghong Peng, UK
 Yu Jian, China
 Yueguo Chen, China
 Zhenying He, China
 Zhihong Deng, China
 Zhongfei Zhang, USA
 Zi Huang, Australia
 Zijiang Yang, Canada

Table of Contents – Part I

Retrieval in CBR Using a Combination of Similarity and Association Knowledge	1
<i>Yong-Bin Kang, Shonali Krishnaswamy, and Arkady Zaslavsky</i>	
A Clustering Approach Using Weighted Similarity Majority Margins . . .	15
<i>Raymond Bisdorff, Patrick Meyer, and Alexandru-Liviu Olteanu</i>	
A False Negative Maximal Frequent Itemset Mining Algorithm over Stream	29
<i>Haifeng Li and Ning Zhang</i>	
A Graph Enrichment Based Clustering over Vertically Partitioned Data	42
<i>Khalid Benabdeslem, Brice Effantin, and Haytham Elghazel</i>	
A Method for Finding Groups of Related Herbs in Traditional Chinese Medicine	55
<i>Lidong Wang, Yin Zhang, Baogang Wei, Jie Yuan, and Xia Ye</i>	
A New Hybrid Clustering Method for Reducing Very Large Spatio-temporal Dataset	69
<i>Michael Whelan, Nhien-An Le-Khac, and M.-Tahar Kechadi</i>	
A Normal Distribution-Based Over-Sampling Approach to Imbalanced Data Classification	83
<i>Huaxiang Zhang and Zhichao Wang</i>	
A Novel Genetic Algorithm for Overlapping Community Detection	97
<i>Yanan Cai, Chuan Shi, Yuxiao Dong, Qing Ke, and Bin Wu</i>	
A Probabilistic Topic Model with Social Tags for Query Reformulation in Informational Search	109
<i>Yuqing Mao, Haifeng Shen, and Chengzheng Sun</i>	
A QoS-Aware Web Services Selection Model Using AND/OR Graph	124
<i>Hong Yu and Man Liu</i>	
A Tweet-Centric Approach for Topic-Specific Author Ranking in Micro-Blog	138
<i>Shoubin Kong and Ling Feng</i>	
An Algorithm for Sample and Data Dimensionality Reduction Using Fast Simulated Annealing	152
<i>Szymon Lukasik and Piotr Kulczycki</i>	

An Investigation of Recursive Auto-associative Memory in Sentiment Detection	162
<i>Saeed Danesh, Wei Liu, Tim French, and Mark Reynolds</i>	
APPECT: An Approximate Backbone-Based Clustering Algorithm for Tags	175
<i>Yu Zong, Guandong Xu, Ping Jin, Yanchun Zhang, EnHong Chen, and Rong Pan</i>	
Bi-clustering Gene Expression Data Using Co-similarity	190
<i>Syed Fawad Hussain</i>	
CCE: A Chinese Concept Encyclopedia Incorporating the Expert-Edited Chinese Concept Dictionary with Online Cyclopedias	201
<i>Jiazhen Nian, Shan Jiang, Congrui Huang, and Yan Zhang</i>	
Cluster Ensembles via Weighted Graph Regularized Nonnegative Matrix Factorization	215
<i>Liang Du, Xuan Li, and Yi-Dong Shen</i>	
Continuously Identifying Representatives Out of Massive Streams	229
<i>Qiong Li, Xiuli Ma, Shiwei Tang, and Shuiyuan Xie</i>	
Cost-Sensitive Decision Tree for Uncertain Data	243
<i>Mingjian Liu, Yang Zhang, Xing Zhang, and Yong Wang</i>	
Direct Marketing with Fewer Mistakes	256
<i>Eileen A. Ni and Charles X. Ling</i>	
Discovering Collective Viewpoints on Micro-blogging Events Based on Community and Temporal Aspects	270
<i>Bin Zhao, Zhao Zhang, Yanhui Gu, Xueqing Gong, Weining Qian, and Aoying Zhou</i>	
Discriminatory Confidence Analysis in Pattern Mining	285
<i>Russel Pears, Yun Sing Koh, and Gillian Dobbie</i>	
Dominance-Based Soft Set Approach in Decision-Making Analysis	299
<i>Awang Mohd Isa, Ahmad Nazari Mohd Rose, and Mustafa Mat Deris</i>	
Efficient Computation of Measurements of Correlated Patterns in Uncertain Data	311
<i>Lisi Chen, Shengfei Shi, and Jing Lv</i>	
Efficient Subject-Oriented Evaluating and Mining Methods for Data with Schema Uncertainty	325
<i>Yue Wang, Changjie Tang, Tengjiao Wang, Dongqing Yang, and Jun Zhu</i>	

An Empirical Evaluation of Bagging with Different Algorithms on Imbalanced Data	339
<i>Guohua Liang and Chengqi Zhang</i>	
Exploiting Concept Clumping for Efficient Incremental News Article Categorization	353
<i>Alfred Krzywicki and Wayne Wobcke</i>	
Extracting Rocks from Mars Images with Data Fields	367
<i>Shuliang Wang and Yashen Chen</i>	
Finding a Wise Group of Experts in Social Networks	381
<i>Hongzhi Yin, Bin Cui, and Yuxin Huang</i>	
Fully Utilize Feedbacks: Language Model Based Relevance Feedback in Information Retrieval	395
<i>Sheng-Long Lv, Zhi-Hong Deng, Hang Yu, Ning Gao, and Jia-Jian Jiang</i>	
FXProj – A Fuzzy XML Documents Projected Clustering Based on Structure and Content	406
<i>Tengfei Ji, Xiaoyuan Bao, and Dongqing Yang</i>	
Author Index	421

Table of Contents – Part II

Generating Syntactic Tree Templates for Feature-Based Opinion Mining	1
<i>Liang Wu, Yuanchun Zhou, Fei Tan, Fenglei Yang, and Jianhui Li</i>	
Handling Concept Drift via Ensemble and Class Distribution Estimation Technique	13
<i>Nachai Limsetto and Kitsana Waiyamai</i>	
HUE-Stream: Evolution-Based Clustering Technique for Heterogeneous Data Streams with Uncertainty	27
<i>Wicha Meesuksabai, Thanapat Kangkachit, and Kitsana Waiyamai</i>	
Hybrid Artificial Immune Algorithm and CMAC Neural Network Classifier for Supporting Business and Medical Decision Making	41
<i>Jui-Yu Wu</i>	
Improving Suffix Tree Clustering with New Ranking and Similarity Measures	55
<i>Phiradit Worawitphinyo, Xiaoying Gao, and Shahida Jabeen</i>	
Individual Doctor Recommendation Model on Medical Social Network	69
<i>Jibing Gong and Shengtao Sun</i>	
Influence Maximizing and Local Influenced Community Detection Based on Multiple Spread Model	82
<i>Qiuling Yan, Shaosong Guo, and Dongqing Yang</i>	
Interactive Predicate Suggestion for Keyword Search on RDF Graphs	96
<i>Mengxia Jiang, Yueguo Chen, Jinchuan Chen, and Xiaoyong Du</i>	
Intrinsic Dimension Induced Similarity Measure for Clustering	110
<i>Yu Xiao, Jian Yu, and Shu Gong</i>	
Learning to Make Social Recommendations: A Model-Based Approach	124
<i>Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, Yang Sok Kim, Paul Compton, and Ashesh Mahidadia</i>	
Microgroup Mining on TSina via Network Structure and User Attribute	138
<i>Xiaobing Xiong, Xiang Niu, Gang Zhou, Ke Xu, and Yongzhong Huang</i>	

Mining Good Sliding Window for Positive Pathogens Prediction in Pathogenic Spectrum Analysis	152
<i>Lei Duan, Changjie Tang, Chi Gou, Min Jiang, and Jie Zuo</i>	
Mining Patterns from Longitudinal Studies	166
<i>Aída Jiménez, Fernando Berzal, and Juan-Carlos Cubero</i>	
Mining Top-K Sequential Rules	180
<i>Philippe Fournier-Viger and Vincent S. Tseng</i>	
Mining Uncertain Data Streams Using Clustering Feature Decision Trees	195
<i>Wenhua Xu, Zheng Qin, Hao Hu, and Nan Zhao</i>	
Multi-view Laplacian Support Vector Machines	209
<i>Shiliang Sun</i>	
New Developments of Determinacy Analysis	223
<i>Rein Kuusik and Grete Lind</i>	
On Mining Anomalous Patterns in Road Traffic Streams	237
<i>Linsey Xiaolin Pang, Sanjay Chawla, Wei Liu, and Yu Zheng</i>	
Ontology Guided Data Linkage Framework for Discovering Meaningful Data Facts	252
<i>Mohammed Gollapalli, Xue Li, Ian Wood, and Guido Governatori</i>	
Predicting New User’s Behavior in Online Dating Systems	266
<i>Tingting Wang, Hongyan Liu, Jun He, Xuan Jiang, and Xiaoyong Du</i>	
Sequential Pattern Mining from Stream Data	278
<i>Adam Koper and Hung Son Nguyen</i>	
Social Influence Modeling on Smartphone Usage	292
<i>Masaji Katagiri and Minoru Etoh</i>	
Social Network Inference of Smartphone Users Based on Information Diffusion Models	304
<i>Tomonobu Ozaki and Minoru Etoh</i>	
Support Vector Regression with A Priori Knowledge Used in Order Execution Strategies Based on VWAP	318
<i>Marcin Orchel</i>	
Terrorist Organization Behavior Prediction Algorithm Based on Context Subspace	332
<i>Anrong Xue, Wei Wang, and Mingcai Zhang</i>	

Topic Discovery and Topic-Driven Clustering for Audit Method Datasets	346
<i>Ying Zhao, Wanyu Fu, and Shaobin Huang</i>	
Transportation Modes Identification from Mobile Phone Data Using Probabilistic Models	359
<i>Dafeng Xu, Guojie Song, Peng Gao, Rongzeng Cao, Xinwei Nie, and Kunqing Xie</i>	
User Graph Regularized Pairwise Matrix Factorization for Item Recommendation	372
<i>Liang Du, Xuan Li, and Yi-Dong Shen</i>	
Using Predicate-Argument Structures for Context-Dependent Opinion Retrieval	386
<i>Sylvester Olubolu Orimaye, Saadat M. Alhashmi, and Siew Eu-Gene</i>	
XML Document Clustering Using Structure-Preserving Flat Representation of XML Content and Structure	403
<i>Fedja Hadzic, Michael Hecker, and Andrea Tagarelli</i>	
Author Index	417

Retrieval in CBR Using a Combination of Similarity and Association Knowledge

Yong-Bin Kang¹, Shonali Krishnaswamy^{1,2}, and Arkady Zaslavsky^{1,3}

¹ Faculty of IT, Monash University, Australia
{yongbin.kang, shonali.krishnaswamy}@monash.edu

² Institute for Infocomm Research (I²R), Singapore

³ Information Engineering Laboratory, ICT Centre, CSIRO, Australia
arkady.zaslavsky@csiro.au

Abstract. Retrieval is often considered the most important phase in Case-Based Reasoning (CBR), since it lays the foundation for the overall performance of CBR systems. In CBR, a typical retrieval strategy is realized through similarity knowledge and is called similarity-based retrieval (SBR). In this paper, we propose and validate that association analysis techniques can be used to enhance SBR. We propose a new retrieval strategy USIMSCAR that achieves the retrieval process in CBR by integrating similarity and association knowledge. We evaluate USIMSCAR, in comparison with SBR, using the Yahoo! Webscope Movie dataset. Through our evaluation, we show that USIMSCAR is an effective retrieval strategy for CBR that strengthens SBR.

1 Introduction

The premise of CBR is that experience in the form of past cases can be leveraged to solve new problems. In CBR, experiences are stored in a database known as a *case base*, and an individual experience is called a *case*. Typically, there are four well-organized phases adopted in CBR [1]: *Retrieve* one or several cases considered useful for solving a given target problem, *Reuse* the solution information of the retrieved cases, *Revise* the solution information to better fit the target problem, and *Retain* the new solution once it has been confirmed or validated.

Retrieval is considered a key phase in CBR, since it lays the foundation for overall performance of CBR systems [2]. Its aim is to retrieve *useful* cases that can be successfully used to solve a new problem. If the retrieved cases are not useful, CBR systems will not eventually produce any good solution for the new problem. To achieve the retrieval process, CBR systems typically rely on a retrieval strategy that exploits *similarity knowledge* and is referred to as *similarity-based retrieval* (SBR) [3]. In SBR, similarity knowledge aims to approximate the *usefulness* of stored cases with respect to the target problem [4]. This knowledge is usually encoded in the form of similarity measures used to compute similarities between a new problem and the cases. By using similarity measures, SBR finds cases with higher similarities to the new problem, and then their solutions are utilized to solve the problem. Thus, it is evident that SBR tends to rely strongly

on similarity knowledge, ignoring other forms of knowledge that can be further leveraged for improving the retrieval performance [3,4,5,6].

In this paper, we propose that association analysis of stored cases can improve traditional SBR. We propose a new retrieval strategy USIMSCAR that leverages *association knowledge* in conjunction with similarity knowledge. Association knowledge is aimed to represent certain interesting relationships, shared by a large number of cases, acquired from stored cases using association rule mining. We show USIMSCAR improves SBR through an experimental evaluation using the ‘Yahoo! Webscope Movie’ dataset. This paper is organized as follows. Section 2 presents our research motivation. Section 3 reviews the related work. Section 3 presents a background of similarity knowledge and association knowledge. Section 4 presents our approach for extracting and representing association knowledge. Section 5 presents the USIMSCAR algorithm. Section 6 evaluates USIMSCAR in comparison with SBR. Section 7 presents our conclusion and future research directions.

2 Motivation

To illustrate our research motivation, we use a medical diagnosis scenario presented in [7]. Consider a case base \mathcal{D} that consists of five patient cases P_1, \dots, P_5 shown in Table 1. Each case is represented by a problem described by 5 attributes (symptoms) A_1, \dots, A_5 , and a corresponding solution described by an attribute (diagnosis) A_6 . Our aim is to determine the correct diagnosis for a new patient Q . We note that Q was suffering from ‘appendicitis’ as specified in [7], and this therefore represents the correct diagnosis.

Table 1. A patient case base

Cases	Local Pain(A_1)	Other Pain(A_2)	Fever (A_3)	Appetite Loss(A_4)	Age (A_5)	Diagnosis (A_6)	Similarity to Q
p_1	right flank	vomit	38.6	yes	10	appendicitis	0.631
p_2	right flank	vomit	38.7	yes	11	appendicitis	0.623
p_3	right flank	vomit	38.8	yes	13	appendicitis	0.618
p_4	right flank	sickness	37.5	yes	35	gastritis	0.637
p_5	epigastrium	nausea	36.8	no	20	stitch	0.420
Q	right flank	nausea	37.8	yes	14	?	
Weight	0.91	0.78	0.60	0.40	0.20		

To predict a diagnosis for Q , SBR retrieves the most similar cases to Q by identifying the cases whose attributes are similar to those of Q using a similarity metric. We use the following metric, the same one used in the work [7], measuring the similarity between Q and each case $p \in \mathcal{D}$,

$$SIM(Q, p) = \frac{\sum_{i=1}^n w_i \cdot sim(q_i, p_i)}{\sum_{i=1}^n w_i},$$

$$sim(q_i, p_i) = \begin{cases} 1 - \frac{|q_i - p_i|}{A_i^{\max} - A_i^{\min}}, & \text{if } A_i \text{ is numeric,} \\ 1, & \text{if } A_i \text{ is discrete \& } q_i = p_i, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where w_i is a weight assigned to an attribute A_i , q_i and p_i are values of A_i of Q and p respectively, n is the number of attributes of Q and p (i.e. $n=5$), $sim(q_i, p_i)$ denotes a similarity measure between q_i and p_i , and A_i^{\max} and A_i^{\min} are the maximum and minimum values, respectively, that A_i takes on. Using the above metric, assume that SBR chooses the most similar case to Q . As seen in Table 1, p_4 is thus chosen, since it is the most similar case to Q . It means that a diagnosis choice for Q is ‘gastritis’. But it turned out to be wrong, since Q suffered from ‘appendicitis’ as mentioned above. To overcome the problem, our idea is to extract, represent, and exploit the knowledge of how known problems are highly associated with known solutions in \mathcal{D} . In \mathcal{D} , we may obtain the knowledge that the problems of cases p_1 , p_2 and p_3 are highly associated with ‘appendicitis’, while those of a case p_4 with ‘gastritis’. The former association strength S_1 may be higher than the latter one S_2 , since S_1 is supported by three cases, while S_2 by a single case. If such strength were to be appropriately quantified, and combined with the similarities in shown Table 1, a diagnosis for Q can be more accurately determined. This is the key idea of our proposed USIMSCAR.

3 Related Work

SBR has been widely used in various CBR application domains, such as medical diagnosis [8] and product recommendation [9], to predict useful cases with respect to the target problem Q . It is typically implemented through *k-nearest neighbor retrieval* or simply *k-NN* [2]. In a CBR context, the idea of *k-NN* is that the retrieval process in CBR is achieved through retrieving the k most similar cases to Q . Thus, the quality of the employed similarity measures for determining those cases is an important aspect in *k-NN*. Over the years, researchers have studied *k-NN* to enhance its accuracy. For example, it is shown that *k-NN* can be integrated with feature selection (FS) [10]. FS is a technique for determining relevant features (or attributes) from the original features of cases. *k-NN* is easily extended to include FS by only considering relevant features when computing the similarity between Q and each case.

To enhance SBR, much work has also focused on integrating SBR with *domain knowledge* and *adaptation knowledge*. For example, Stahl [4] proposes a retrieval approach in which similarity assessment during SBR is integrated with domain knowledge. Aamodt [11] proposes an approach that cases are enriched with domain knowledge that guides the retrieval of relevant cases. Adaptation knowledge is also used to enhance SBR in which this knowledge indicates whether a case can be easily modified to fit the new problem [3]. In this approach, matches between the target problem and cases are done, only if there is evidence in adaptation knowledge that such matches can be catered for during retrieval.

Our approach for enhancing SBR differs from the above approaches in three aspects: (1) While many kinds of learnt and induced knowledge has been utilized, we leverage association knowledge that has not been used for retrieval in CBR systems. (2) The acquisition of both domain and adaptation knowledge is usually known as a very complex and difficult task, thus often leads to knowledge bottleneck phenomenon [4]. However, association knowledge acquisition is straightforward, since

it is automatically acquired from stored cases, a fundamental knowledge source in CBR, using association rule mining. (3) Association knowledge extraction is achieved through capturing strongly evident associations between known problem features and solutions shared by a large number of cases. This scheme can be compared to FS, since in a CBR context it mainly focuses on estimating the relevance of problem features highly correlated to known solutions. However, FS usually assumes feature independence, ignoring identifying interesting relationships between problem features, dependent on each other, and each solution. In contrast, association knowledge extraction includes and considers all interesting frequent patterns and association structures from a given case base using association rule mining.

4 Background of Similarity and Association Knowledge

Prior to presenting our proposed USIMSCAR, we provide a background of similarity and association knowledge. We first present our case representation scheme that is the basis for representing both similarity and association knowledge. To represent cases, many CBR systems generally adopt well-known knowledge representation formalisms, such as *attribute-value pairs* and *structural* representations [4]. In our work, we choose the attribute-value pairs representation due to its simplicity, flexibility and popularity. Let A_1, \dots, A_m be attributes defined in a given domain. An *attribute-value pair* is a pair (A_i, a_i) , where A_i is an attribute (or feature¹) and a_i is a value of $A_{i \in [1, m]}$. A *case* C is a pair $C = (X, Y)$ where X is a problem, represented as $X = \{(A_1, a_1), \dots, (A_{m-1}, a_{m-1})\}$, and Y is the solution of X , represented as $Y = (A_m, a_m)$. We call an attribute A_m a *solution-attribute*. A *case base* \mathcal{D} is a collection of cases.

4.1 Background of Similarity Knowledge

In a CBR context, we refer to similarity knowledge as knowledge encoded via measures computing the similarities between the target problem and stored cases. To formulate the measures, CBR systems often use a widely used principle. This is the *local-global principle* that decomposes a similarity measure by *local similarities* for individual attributes, and a *global similarity* aggregating the local similarities [4]. An accurate definition of local similarities relies on attribute types. A global similarity function can be arbitrarily complex, but usually simple functions (e.g. weighted average) are used in many CBR systems. Referring to Eq. 1 *SIM* is a global similarity function, and *sim* is a local similarity function.

4.2 Background of Association Knowledge

Our premise is that SBR can be enhanced by the inclusion of association knowledge representing evidently interesting relationships shared by a large number of stored cases. It is extracted from stored cases and represented using *association*

¹ To simplify the presentation, we do not distinguish between terms “attributes” and “features”, and use these terms interchangeably.

rule mining [12], class association rule mining [13] and soft-matching criterion [14], which are outlined in the following.

Association rule mining [12] aims to mine certain interesting associations in a transaction database. Let I be a set of distinct literals called *items*. A set of items $X \subseteq I$ is called an itemset. Let \mathcal{D} be a set of transactions. Each transaction $T \in \mathcal{D}$ is a set of items such that $T \subseteq I$. We say that T *contains* an itemset X , if $X \subseteq T$ holds. Every *association rule* has two parts: an *antecedent* and a *consequent*. An association rule is an implication of the form $X \rightarrow Y$, where $X \subseteq I$ is an itemset in the antecedent and $Y \subseteq I$ is an itemset in the consequent, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has *support* s in \mathcal{D} if $s\%$ of transactions in \mathcal{D} contain $X \cup Y$. This holds in \mathcal{D} with *confidence* c if $c\%$ of transactions in \mathcal{D} that contain X also contain Y . Association rule mining can also be used for discovering interesting relationships among stored cases. In a CBR context, a transaction can be seen as a case, and an item as an attribute-value pair. Referring to Table 1, we can mine a rule $r_1 : (A_1, \text{right flank}) \rightarrow (A_2, \text{vomit})$. Let X be an item $(A_1, \text{right flank})$. Let Y be an item (A_2, vomit) . The support of r_1 is 0.6, since X and Y occur together in three out of five cases in \mathcal{D} . The confidence of r_1 is 0.75, since Y occurs in three out of four cases that contain X in \mathcal{D} . Apriori [12] is one of the traditional algorithms for association rule mining.

Class association rules (cars) [13] is a special subset of association rules whose consequents are restricted to a single target variable. In a CBR context, cars can be seen as special association rules whose consequents only hold special items formed as pairs of a “solution-attribute” and its values. We call such an item a *solution-item*. Thus, a car has the form $X \rightarrow y$, where $X \subseteq I$ an itemset in the antecedent and $y \in I$ is a solution-item in the consequent. Our aim of building association knowledge is to represent the knowledge encoding how certain known problems are associated with known solutions in a case base. For the purpose, we use the form of cars, since it is suited for this goal. Note that the car $X \rightarrow y$ encodes an association between an itemset X (i.e. attribute-value pairs of known problems), and a solution-item y (i.e. the corresponding solution).

Consider the association rule $X \rightarrow Y$. A limitation of traditional association rule mining algorithms (e.g. Apriori [12]) is that itemsets X and Y are discovered using equality relation. Unfortunately, when dealing with items similar to each other, these algorithms may perform poorly. For example, a supermarket sales database, Apriori cannot find rules like “80% of the customers who buy products similar to milk (e.g. cheese) and products similar to eggs (e.g. mayonnaise) also buy bread.” To address this issue, SoftApriori [14] was proposed. It uses the *soft-matching criterion*, where the antecedent and the consequent of association rules are found using similarity assessment. By doing so, this criterion can be used to model richer relationships among case features than the equality relation.

5 Association Knowledge Formalization

This section presents our approach for extracting and representing association knowledge used the techniques outlined in Section 3. The aim of building association knowledge is two-fold. The first is to represent strongly evident, interesting

associations between known problem features and solutions shared by a large number of cases. The second is to leverage these associations along with similarity knowledge in our proposed USIMSCAR to improve SBR.

We propose to represent association knowledge via cars whose antecedents are determined by applying the soft-matching criterion. We refer to these rules as *soft-matching class association rules* (scars). A scar $X \rightarrow y$ implies that the target problem Q is likely to be associated with the solution contained in an item y , if the problem features of Q are highly similar to an itemset X .

Let \mathcal{D} be a set of cases, where each case is characterized by attributes A_1, \dots, A_m . We call a pair $(A_i, a_i)_{i \in [1, m-1]}$ an *item*. We call a pair (A_m, a_m) a *solution-item*. Let I be a set of items. A set $L \subseteq I$ with $k = |L|$ is called a k -itemset or simply an itemset. Let $\text{sim}(x, y)$ be a function computing the similarity between two items $x, y \in I$ in terms of their values. We say that x and y are similar, iff $\text{sim}(x, y) \geq a$ *user-specified minimum similarity* (minsim). Given two itemsets $X, Y \subseteq I$ ($|X| \leq |Y|$), $\text{ASIM}(X, Y)$ is a function that computes the asymmetric similarity of X with respect to Y , defined as $\sum_{x \in X, y \in Y} \frac{\text{sim}(x, y)}{|X|}$, where x, y are items with the *same* attribute label. Let X_1 be a 2-itemset $\{(A_1, a), (A_2, b)\}$. Let Y_1 be a 2-itemset $\{(A_1, a'), (A_2, b')\}$. Assuming similarity functions for A_1 and A_2 are denoted as sim_{A_1} and sim_{A_2} respectively, $\text{ASIM}(X_1, Y_1) = (\text{sim}_{A_1}(a, a') + \text{sim}_{A_2}(b, b'))/2$. We say that X is a soft-subset of Y ($X \subseteq_{\text{soft}} Y$), iff $\text{ASIM}(X, Y) \geq \text{minsim}$; or Y *softly contains* X . The *soft-support-sum* of an itemset $X \subseteq I$ is defined as the sum of the asymmetric similarities of X with respect to cases in \mathcal{D} that softly contain X , $\text{softSuppSum}(X) = \sum_{X \subseteq_{\text{soft}} C \in \mathcal{D}} \text{ASIM}(X, C)$. The *soft-support* of X is defined as $\text{softSupp}(X) = \text{softSuppSum}(X)/|\mathcal{D}|$. The *soft-support-sum* of a rule $X \rightarrow y$ is defined as the sum of the asymmetric similarities of X with respect to cases in \mathcal{D} that softly contain X and contain y , $\text{softSuppSum}(X \rightarrow y)$. The *soft-support* of this rule is defined as $\text{softSupp}(X \rightarrow y) = \text{softSuppSum}(X \rightarrow y)/|\mathcal{D}|$. The *soft-confidence-sum* of a rule $X \rightarrow y$ is defined as the sum of the asymmetric similarities of X with respect to cases in \mathcal{D} that softly contain X also contain y , $\text{softConfSum}(X \rightarrow y)$. The *soft-confidence* of this rule is defined as $\text{softConf}(X \rightarrow y) = \text{softConfSum}(X \rightarrow y)/|\mathcal{D}|$.

The key operation for scars mining is to find all ruleitems that have soft-supports $\geq (a \text{ user-specified minimum support})$ (minsupp). We call such ruleitems *frequent* ruleitems. For all the ruleitems that have the same itemset in the antecedent, one with the highest *interestingness* is chosen as a *possible rule* (PR). To measure the interestingness of association rules, support and confidence are typically used. On some occasions, a combination of them is used. Often, a rationale for doing so is to define a single optimal interestingness by leveraging their correlations. We choose the Laplace measure (LM) [15] that combines soft-support and soft-confidence such that they are monotonically related. Given a ruleitem $r : X \rightarrow y$, its LM *Laplace*(r) can be denoted as $\frac{|\mathcal{D}| \cdot \text{softSupp}(X \rightarrow y) + 1}{|\mathcal{D}| \cdot \text{softSupp}(X \rightarrow y) / \text{softConf}(X \rightarrow y) + 2}$. If *Laplace*(r) $\geq a$ *user-specified minimum level of interesting* (min-interesting), we say r is *accurate*. A candidate set of scars consists of all the PRs that are frequent and accurate.

Let k -ruleitem be a ruleitem whose antecedent has k items. Let F_k be a set of frequent k -ruleitems. The following is a description for the scars mining algorithm: (1) For 1-ruleitems $X \subseteq I$, we find $F_1 = \{\{X\} | \text{softSupp}(X) \geq \text{minsupp}\}$. A set $SCAR_1$ is then generated by only choosing PRs from F_1 . (2) For each k subsequent pass, we find a set of new possibly frequent ruleitems CR_k using F_{k-1} found in the $(k-1)^{th}$ pass. We then generate a new set F_k by extracting ruleitems in CR_k whose soft-support $\geq \text{minsupp}$. A set $SCAR_k$ is generated by only choosing PRs from F_k . (3) From $SCAR_1, \dots, SCAR_k$, we choose only sets whose $i \in [1, k] \geq a$ user-specified minimum ruleitem size (minitemsize), and store them in a set $SCARS$. Our idea is to choose a small representative subset of frequent ruleitems from the large number of resulting frequent ruleitems. The longer the frequent ruleitem, the more significant it is [16]. We perform a rule pruning on ruleitems in $SCARS$. A rule r is pruned, if $Laplace(r) < \text{min-interesting}$. The set of ruleitems after the pruning is finally returned as the set of scars to be used in our proposed USIMSCAR.

6 The USIMSCAR Algorithm

This section presents USIMSCAR that leverages both association and similarity knowledge to enhance SBR. The main challenge is how to combine similarity and association knowledge appropriately and effectively, thereby strengthening the retrieval performance of SBR. This section address this challenge by presenting the USIMSCAR algorithm. The rationale for leveraging association knowledge in USIMSCAR falls into two objectives: (1) enhancing the usefulness of the cases, retrieved by using similarity knowledge as with SBR, with respect to a new problem Q by including both similarity and association knowledge, and (2) directly leveraging a number of scars whose usefulness is relatively high with respect to Q , eventually utilizing such scars with their usefulness in USIMSCAR.

Given a new problem Q , USIMSCAR's goal is to produce a retrieval result RR consisting of objects that can be used to solve Q by leveraging similarity and association knowledge. Such objects are obtained from both stored cases and scars mined. Let \mathcal{D} be a set of cases. Let $SCARS$ be the set of scars mined from \mathcal{D} . Below we present the USIMSCAR algorithm.

(1) From \mathcal{D} , we find the k most similar cases to Q , and store them in a set RC . We denote $SIM(Q, C)$ as the similarity between Q and a case C .

(2) In $SCARS$, we find the k' most similar scars to Q , and store them in a set RS . A question raised here is how to compute the similarity $SIM(Q, r)$ between Q and a scar r . Its answer lies in our choice of cars representation for scars mining. Note that scars have the *identical* structure as cases: the antecedent and consequent correspond to the problem and solution part of cases respectively. Thus, $SIM(r, Q)$ can be defined in the same way as $SIM(Q, C)$ in (1). To generate RS , we only consider scars (RCS) in $SCARS$ such that their antecedents are soft-subsets of cases in RC , rather than all scars in $SCARS$ for efficiency. Each case $C \in RC$ is chosen as a similar case to Q ($C \sim Q$). Assuming each scar $r \in RCS$ has the form $r : X \rightarrow y$, X is a soft-subset of C ($X \subseteq_{\text{soft}} C$). Since

$C \sim Q$ and $X \subseteq_{soft} C$, $X \subseteq_{soft} C \sim Q$ can be derived. It implies that RCS is a particular subset (i.e. soft-subset) of cases in RC similar to Q .

(3) For each case $C \in RC$, we select a scar $r_C \in SCARS$. It is chosen if it has the highest interestingness among those scars in $SCARS$ such that their antecedents are soft-subsets of C and their consequents are equal to the solution of C . We then quantify the usefulness of C with respect to Q ($USF(C, Q)$) by $SIM(C, Q) \times Laplace(r_C)$. If candidates for r_C are chosen more than one, say m , we use the average of the interestingness of these m scars to compute $Laplace(r_C)$. If there is no candidate for r_C , we use min-interesting for $Laplace(r_C)$. Note that in SBR, the usefulness of C regarding Q is measured by $SIM(C, Q)$. Our combination schemes aims to quantify this usefulness by leveraging $SIM(C, Q)$ and $Laplace(r_C)$. We then cast C to a generic object O that can hold any cases and scars. O has two fields: $O.inst = C$, $O.usf = USF(C, Q)$. The object O is then added to a retrieval result RR .

(4) For each scar $r \in RS$, we quantify the usefulness of r with respect to Q ($USF(r, Q)$) by $SIM(r, Q) \times Laplace(r_C)$. This aims to quantify the usefulness by combining $SIM(r, Q)$ obtained from similarity knowledge and $Laplace(r_C)$ acquired from association knowledge. We then cast r to a generic object O with two fields: $O.inst = r$, $O.usf = USF(r, Q)$. The object O is then added to RR .

(5) We further enhance the usefulness of each object $O \in RR$ using the frequency of *solution occurrence* among objects in RR . Our premise is that if O 's solution is more frequent in RR , O is more useful in RR . If O is cast from a case C , its solution means C 's solution. If O cast from a scar r , its solution is r 's consequent. Let S be a set of solutions of objects in RR . Let S_O be a set of objects in RR that have the solution equal to the solution of an object $O \in RR$. For each object $O \in RR$, we compute $\delta(S_O)$ as $\delta(S_O) = |S_O|/|RR|$. Finally, we enhance $O.usf$ by multiplying $\delta(S_O)$. Eventually, each object $O \in RR$ with $O.usf$ is utilized to induce a solution for Q .

We now illustrate how USIMSCAR operates using the case base \mathcal{D} shown in Table 1. From \mathcal{D} , we can generate 4 scars shown in Table 2 using the similarity SIM in Eq. 1.

Table 2. The scars generated

Rules	Laplace	Soft-subset of
$r_1: \{(A_1, \text{right flank}), (A_2, \text{vomit}), (A_3, 38.6), (A_4, \text{yes}), (A_5, 13)\} \rightarrow (A_6, \text{appendicitis})$	0.922	p_1, p_2, p_3
$r_2: \{(A_1, \text{right flank}), (A_2, \text{vomit}), (A_3, 38.7), (A_4, \text{yes}), (A_5, 10)\} \rightarrow (A_6, \text{appendicitis})$	0.922	p_1, p_2, p_3
$r_3: \{(A_1, \text{right flank}), (A_2, \text{vomit}), (A_3, 38.8), (A_4, \text{yes}), (A_5, 10)\} \rightarrow (A_6, \text{appendicitis})$	0.922	p_1, p_2, p_3
$r_4: \{(A_1, \text{right flank}), (A_2, \text{sickness}), (A_3, 37.5), (A_4, \text{yes}), (A_5, 35)\} \rightarrow (A_6, \text{gastritis})$	0.775	p_4

Using the above scars, USIMSCAR takes the following steps (assume $k=k'=2$): (1) It finds the 2 most similar cases to Q : $RC = \{p_4, p_1\}$ for $SIM(Q, p_4)=0.637$, $SIM(Q, p_1)=0.631$. (2) It finds the 2 most similar scars to Q : $RS = \{r_1, r_4\}$ for $SIM(Q, r_1)=0.640$, $SIM(Q, r_4)=0.637$. (3) For each case $C \in RC$, r_C is chosen. For p_4 , r_4 is selected. For p_1 , r_1 , r_2 and r_3 are selected. Then, $USF(Q, p_4)$ and $USF(Q, p_1)$ are quantified as $USF(Q, p_4)=0.494$, $USF(Q, p_1)=0.581$. Then, p_4 with $USF(Q, p_4)$ and p_1 with $USF(Q, p_1)$ are cast to new objects and stored

in a set RR . (4) For each scar $r \in RS$, its usefulness to Q is quantified as $USF(Q, r_1)=0.594$, $USF(Q, r_4)=0.496$. Then, these scars with their usefulness are cast to new objects and stored in RR . (5) Assume that each object in RR has another field s holding its solution. RR has 4 objects $RR = \{O_1, \dots, O_4\}$ shown in Table 3. As observed, there are only two sets of objects regarding solutions. For each object $O \in RR$, $O.usf$ is enhanced by weighting $\delta(S_O) = |S_O|/|RR|$. The enhancement results are shown under the column ‘final usf’ in the table. Eventually, if we choose the most useful one to Q , we retrieve O_3 and its solution ‘appendicitis’, Q really had, is used as a diagnosis for Q .

Table 3. The retrieval result RR

field: inst	field: usf	field: solution	final usf
$O_1.inst = p_4$,	$O_1.usf = 0.494$,	$O_1.s =$ gastritis	0.247
$O_2.inst = p_1$,	$O_2.usf = 0.581$,	$O_2.s =$ appendicitis	0.291
$O_3.inst = r_1$,	$O_3.usf = 0.594$,	$O_3.s =$ appendicitis	0.297
$O_4.inst = r_4$,	$O_4.usf = 0.496$,	$O_4.s =$ gastritis	0.248

7 Evaluation

We experimentally show that USIMSCAR improves SBR with respect to retrieval performance. Our work has focused on proposing a new retrieval strategy for CBR. Thus, as a target application task, it is desirable to choose a task that is highly dependent on retrieval performance in a CBR context. One suitable task is case-based classification [17], defined as: given a new problem Q , its goal is to find similar cases to Q from a case base, and classify Q based on the retrieved cases. Thus, in principle, this approach is strongly dependent on the result obtained through retrieval in CBR.

As target SBR approaches to be compared with USIMSCAR, we choose the following k -NN approaches implemented in Weka, since SBR is typically implemented through k -NN: (1) IB1 is the simplest form of k -NN using the Euclidean distance to find the most similar case C to Q . (2) IBkBN extends IB1 by using the best k (i.e. the number of the most similar cases) determined by cross-validation. (3) IBkFS extends IBkBN by using a feature selector CfsSubsetEval available in Weka. (4) KStar is an implementation of K^* [18], where similarity for finding the most similar cases to Q is defined through entropy.

In a k -NN approaches context, classification has two stages. The first is to find similar cases RR to Q using similarity knowledge, and the second is to classify Q using the solutions in RR . In a USIMSCAR context, the first is to find a set of ‘‘useful cases and rules’’ RR using ‘‘similarity and association knowledge’’, and the second is to classify Q using the solutions of objects in RR . Our work is focused on the first stage. The second stage can be achieved using *voting*. Due to generality, we adopt *weighted voting*, where objects in RR get to vote, on the solution of Q , with votes weighted by their significance to Q . For each object in RR , in SBR the significance is measured using its similarity to Q , while in USIMSCAR it is measured using its usefulness with respect to Q .

We use the Yahoo! Webscope Movie dataset (R4) usually used for evaluating recommender systems. In a CBR context, each instance in R4 has the form (x, s_x) : x is a problem description characterized by two user attributes (birthyear, gender) and ten movie attributes (see Table 4), and s_x is the corresponding solution meaning a rating assigned to a movie by a user. Before testing, we removed the instances that contain any missing values of any movie attributes, and redundant movie attributes (e.g. actors are represented using both ‘actor id’ and ‘name’, so we included only the name). Finally, R4 consists of training data (74,407 ratings scaled from 1 to 5 rated by 5,826 users for 312 movies), and testing data (2,705 ratings scaled from 1 to 5 rated by 993 users for 262 movies).

Table 4. Movie information (movie-info)

Attributes	Description	Type
title	movie title	String
synopsis	movie synopsis	String
mpaa_rating	MPAA rating of movie	Nominal
genres	list of the genres of movie	Set-valued
directors	list of the directors of movie	Set-valued
actors	list of the actors of movie	Set-valued
avg-critic-rating	average of the critic reviews of movie	Numeric
rating-from-Mom	rating to movies obtained from the Movie Mom	Numeric
gnpp	Global Non-Personalized Popularity (GNPP), of movie, computed by Yahoo! Research	Numeric
avg-rating	average movie rating by users in the training data	Numeric

For each instance Q in the testing data, our goal is to predict the correct rating that the user will be likely to rate using the training data. We split it into three classification tasks taking a user and a movie, and classify a rating in three rating-scales: RS(5) is a five rating-scale [1,5], RS(3) is a 3 rating-scale where a rating indicates whether a movie would be *liked* (> 3), *normal* ($=3$) or *disliked* (<3), and RS(2) is a 2 rating-scale where a rating indicates whether a movie would be *liked* (>3) or *disliked* (≤ 3). We evaluate the prediction using *classification accuracy* (CA) and *predictive accuracy* (PA) that are widely used for classification and recommendations. CA measures the proportion of correctly classified instances over all the instances tested. PA measures how close predicted ratings are to the actual user ratings. *Mean absolute error* (MAE) is widely used to measure this accuracy, $\sum_{i=1}^N |p_i - r_i|/N$, where p_i is a predicted rating, r_i is an actual rating for an instance i , and N is the number of instances tested. Regarding MAE, lower values are more accurate. We compute the MAE values for each user in the testing data, and then average over all users in the data.

The similarity knowledge used is encoded as a similarity measure using the global-local principle. Given a new problem Q and a case, their global similarity is defined as *SIM* in Eq. 1, and local similarities are defined on four types. For numeric and nominal attributes, we used *sim* in Eq. 1. For set-valued attributes, we used the *Jaccard coefficient*. For string attributes, we converted a given string into a set-valued representation by tokenizing it, and applied the Jaccard coefficient. We implemented IBkBN, IBkFS and USIMSCAR to be working with *SIM*

to find the k most similar cases for Q . The function SIM is also used to find the k' most similar scars with respect to Q in USIMSCAR. For the approaches, we chose a best value for k using cross-validation from 1 to 15. We observed that increasing k beyond 15 hardly changed the results.

To generate scars from R4, we set `minsim`, `min-interesting`, and `minitemsize` to arbitrary values 0.95 (95%), 0.7, and 7 respectively. Setting a value for `minsupp` is more complex, since it has a stronger effect on the quality of USIMSCAR. If `minsupp` is set too high, those possible scars, which cannot satisfy `minsupp` but with high interestingness (Laplace measure) values, will not be included. While if `minsupp` is set too low, it is possible to generate too many scars including trivial rules. Both occasions may lead to a reduction in the performance of USIMSCAR. From our experiments, we observed that once `minsupp` is set to 0.1, the performance of USIMSCAR is best. We thus set a value for `minsupp` to 0.1.

7.1 Results and Analysis

We now present the experimental results of USIMSCAR and the compared k -NN approaches (simply 4KNN) in terms of both classification accuracy (CA) and MAE in Tables 5 and 6. For each rating-scale, the best accuracy is denoted in boldface. The mark “★” indicates that USIMSCAR attains a significant improvement over the target measure. For CA, it is discovered by the Z -test [19] with 95% confidence, and for MAE by the paired t -tests [19] at 95% confidence. Each number in parentheses denotes the improvement ratio of USIMSCAR over the target measure.

Table 5 indicates that USIMSCAR achieves 100% better performance than 4KNN in all rating-scales in terms of CA. We find that the 91.6% comparisons between USIMSCAR and 4KNN are statistically significant in terms of CA.

Table 5. The classification accuracy results

Compared Classifiers	Classification Accuracy(%)		
	RS(5)	RS(3)	RS(2)
IB1	46.30 (2.76% ★)	72.95 (6.61% ★)	75.24 (5.02% ★)
IBkBN	48.61 (0.45%)	75.97 (3.59% ★)	77.12 (3.14% ★)
IBkFS	46.28 (2.78% ★)	74.17 (5.39% ★)	75.24 (5.02% ★)
KStar	44.34 (4.72% ★)	74.37 (5.19% ★)	75.07 (5.19% ★)
USIMSCAR	49.06	79.56	80.26

Table 6. The MAE results

Compared Classifiers	Predictive Accuracy (MAE)		
	RS(5)	RS(3)	RS(2)
IB1	.9139 (24.14% ★)	.3760 (8.76% ★)	.2476 (5.02% ★)
IBkBN	.8532 (18.07% ★)	.3482 (5.98% ★)	.2288 (3.08% ★)
IBkFS	.8392 (16.67% ★)	.3541 (6.57% ★)	.2376 (4.02% ★)
KStar	.8710 (19.89% ★)	.3652 (4.02% ★)	.2493 (5.19% ★)
USIMSCAR	.6725	.2884	.1974

As shown in Table 6, we also find that USIMSCAR achieves 100% better performance than 4KNN in all rating-scales in terms of MAE. All the improvements are deemed to be statistically significant. Through these results, we demonstrate that USIMSCAR has the ability to retrieve more useful objects (i.e. cases and scars) with respect to the target problems than SBR. As outlined in the USIMSCAR algorithm, these objects are identified and quantified by using a combination of similarity and association knowledge. This further establishes the validity of the primary motivation of this research that the combination will lead to improving SBR. The real strength of our evaluation lies in the fact that USIMSCAR improves SBR for CBR classification using a real-world dataset.

Up to now, we have formalized the recommendation problem as a classification problem and shown the improvement of USIMSCAR over k -NN classifiers in terms of CA and MAE. In a certain context, it is also important to compare USIMSCAR and existing recommenders. Recommenders are usually classified as follows: content-based (CB) recommenders recommend items similar to the ones that the user has liked in the past, collaborative filtering (CF) recommenders recommend items that other users with similar preferences have liked in the past, and hybrid recommenders recommend items by combining the above two approaches. We see that USIMSCAR is also a unifying model realizing a hybrid recommendation. It differs from CF recommenders in that it exploits content information of items (movies) with rating information. It also differs from CB recommenders by using other users' ratings when building and exploiting association knowledge for rating classification. We compare USIMSCAR with two well-known hybrid recommenders: CLAYPOOL [20] and MELVILLE [21]. For CLAYPOOL, we first applied the CF method proposed by [20] to the training data. We then applied a CB method using IBk to the data. The ratings returned by these methods were combined by the equal-weighted average to produce a final rating. MELVILLE uses a CB method to convert a *sparse* user-ratings matrix UM into a *full* user-ratings matrix FUM. Given a user, a rating prediction is made for a new item using a CF method on the FUM. As the CB predictor, we used IBk. For the CF method, we implemented the algorithm in [21].

The comparison results are seen in Tables 7 and 8. As seen in Table 7, USIMSCAR outperforms the recommenders in all rating-scales in terms of CA. We discover that 50% of comparisons between USIMSCAR and the recommenders are deemed to be statistically significant through the Z -test at 95% confidence. As seen in Table 8, USIMSCAR also outperforms both recommenders in all rating-scales in terms of MAE. We discover that 50% of comparisons between USIMSCAR and the recommenders are also deemed to be statistically

Table 7. The classification results

Recommenders	Classification Accuracy (%)		
	RS(5)	RS(3)	RS(2)
CLAYPOOL	48.95 (0.11%)	77.97 (0.22%)	80.04 (1.59%)
MELVILLE	43.96 (5.10% ★)	73.57 (4.40% ★)	75.86 (5.99% ★)
USIMSCAR	49.06	79.56	80.26

Table 8. The MAE results

Recommenders	Predictive Accuracy (MAE)		
	RS(5)	RS(3)	RS(2)
CLAYPOOL	.6954 (2.29%)	.3102 (1.28%)	.1996 (0.22%)
MELVILLE	.7863 (11.38% ★)	.3579 (6.95% ★)	.2414 (4.40% ★)
USIMSCAR	.6725	.2884	.1974

significant by the paired t -test with 95% confidence. In summary, through all the experiments, we have demonstrated the validity and soundness of our proposed USIMSCAR approach.

8 Conclusion and Future Work

This paper presented a novel retrieval strategy USIMSCAR that can be used in retrieving useful cases for the target problem. First, we proposed an approach for extracting and representing association knowledge that represents strongly evident, interesting associations between known problem features and solutions shared by a large number of cases. We proposed that this knowledge is encoded via soft-matching class association rules (scars) using association analysis techniques. We proposed USIMSCAR that leverages useful cases and rules, with respect to the target problem, quantified by using both similarity and association knowledge. This idea to leveraging the combined knowledge during CBR retrieval clearly distinguishes USIMSCAR from SBR as well as existing retrieval strategies developed in the CBR research community. We validated the improvement of USIMSCAR over well-known k -NN approaches for implementing SBR through experiments using the Yahoo! Webscope Movie dataset. The experimental results showed that USIMSCAR is an effective retrieval strategy for CBR.

In CBR, cases can also be represented by more complex structures, like object-oriented representation (OO) or hierarchical representation (HR) [2]. OO utilizes the data modeling approach of the OO paradigm, such as inheritance. In HR, a case is characterized through multiple levels of abstraction, and its attribute values can reference nonatomic cases [2]. To support these representations, USIMSCAR must address how to generate similarity knowledge and association knowledge. To address the former, one may use similarity measures proposed by [22] for OO data or HR data. To address the latter, one may integrate the soft-matching criterion and extended Apriori algorithms such as ORFP [23] for OO data and DFMLA [24] for HR data.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7, 39–59 (1994)
2. Lopez De Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. *Knowl. Eng. Rev.* 20, 215–240 (2005)

3. Smyth, B., Keane, M.T.: Adaptation-guided retrieval: questioning the similarity assumption in reasoning. *Artif. Intell.* 102, 249–293 (1998)
4. Stahl, A.: Learning of knowledge-intensive similarity measures in case-based reasoning. PhD thesis, Technical University of Kaiserslautern (2003)
5. Cercone, N., An, A., Chan, C.: Rule-induction and case-based reasoning: hybrid architectures appear advantageous. *IEEE Trans. on Know. and Data Eng.* 11, 166–174 (1999)
6. Park, Y.J., Kim, B.C., Chun, S.H.: New knowledge extraction technique using probability for case-based reasoning: application to medical diagnosis. *Expert Systems* 23, 2–20 (2006)
7. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: Loss and gain functions for CBR retrieval. *Inf. Sci.* 179, 1738–1750 (2009)
8. Ahn, H., Kim, K.J.: Global optimization of case-based reasoning for breast cytology diagnosis. *Expert Syst. Appl.* 36, 724–734 (2009)
9. Bradley, K., Smyth, B.: Personalized information ordering: a case study in online recruitment. *Knowledge-Based Systems* 16, 269–275 (2003)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157–1182 (2003)
11. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning in Creek. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004. LNCS (LNAI)*, vol. 3155, pp. 793–850. Springer, Heidelberg (2004)
12. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: *SIGMOD 1993*, pp. 207–216. ACM (1993)
13. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: *Proceedings of the 4th KDD*, pp. 443–447 (1998)
14. Nahm, U.Y., Mooney, R.J.: Mining soft-matching association rules. In: *Proceedings of CIKM 2002*, pp. 681–683 (2002)
15. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. *ACM Comput. Surv.* 38, 9 (2006)
16. Hu, T., Sung, S.Y., Xiong, H., Fu, Q.: Discovery of maximum length frequent itemsets. *Inf. Sci.* 178, 69–87 (2008)
17. Jurisica, I., Glasgow, J.: Case-Based Classification Using Similarity-Based Retrieval. In: *International Conference on Tools with Artificial Intelligence*, p. 410 (1996)
18. Cleary, J.G., Trigg, L.E.: K*: An Instance-based Learner Using an Entropic Distance Measure. In: *Proceedings of the 12th ICML*, pp. 108–114 (1995)
19. Richard, C.S.: *Basic Statistical Analysis*. Allyn & Bacon (2003)
20. Claypool, M., Gokhale, A., Miranda, T.: Combining content-based and collaborative filters in an online newspaper. In: *Proceedings of ACM-SIGIR Workshop on Recommender Systems* (1999)
21. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: *AAAI 2002*, pp. 187–192 (2002)
22. Bergmann, R., Stahl, A.: Similarity measures for object-oriented case representations. In: Smyth, B., Cunningham, P. (eds.) *EWCBR 1998. LNCS (LNAI)*, vol. 1488, pp. 25–36. Springer, Heidelberg (1998)
23. Kuba, P., Popelinsky, L.: Mining frequent patterns in object-oriented data. In: *Technical Report: Masaryk University Brno, Czech Republic* (2005)
24. Pater, S.M., Popescu, D.E.: Market-Basket Problem Solved With Depth First Multi-Level Apriori Mining Algorithm. In: *3rd International Workshop on Soft Computing Applications SOFA 2009*, pp. 133–138 (2009)

A Clustering Approach Using Weighted Similarity Majority Margins

Raymond Bisdorff¹, Patrick Meyer^{2,3}, and Alexandru-Liviu Olteanu^{1,2,3}

¹ CSC/ILIAS, FSTC, University of Luxembourg

² Institut Télécom, Télécom Bretagne, UMR CNRS 3192 Lab-STICC, Technopôle Brest Iroise, CS 83818, 29238 Brest Cedex 3, France

³ Université Européenne de Bretagne

Abstract. We propose a meta-heuristic algorithm for clustering objects that are described on multiple incommensurable attributes defined on different scale types. We make use of a bipolar-valued dual similarity-dissimilarity relation and perform the clustering process by first finding a set of cluster cores and then building a final partition by adding the objects left out to a core in a way which best fits the initial bipolar-valued similarity relation.

1 Introduction

Clustering is defined as the unsupervised process of grouping objects that are similar and separating those that are not. Unlike classification, clustering has no a priori information regarding the groups to which to assign the objects. It is widely used in many fields like artificial intelligence, information technology, image processing, biology, psychology, marketing and others. Due to the large range of applications and different requirements many clustering algorithms have been developed. Jain [16] gives a thorough presentation of many clustering methods and classifies them into partitioning [21,20], hierarchical [13,15,29], density-based [3,30], grid-based [2,27] and model-based methods [9,19]. New graph-based methods have also been developed in the emerging field of community detection [10,25,26]. Fortunato [11] covers many of the latest ones.

In this paper we present the GAMMA-S method (a Grouping Approach using weighted Majority MArgins on Similarities) for clustering objects that are described by multiple incommensurable attributes on nominal, ordinal and/or cardinal scales. We draw inspiration from the bipolar-valued outranking approach proposed by [5,6,7] for dealing with multiple criteria decision aid problems. As such, we assume that the data is extracted in a prior stage, such that each attribute has a clear meaning and expresses a distinct viewpoint for a human agent. Also, this agent has a clear view on the importance of each attribute when he compares two objects and what can be considered as a discriminating difference in their evaluations. For this we first characterize pairwise global similarity statements by balancing marginal similarity and dissimilarity situations observed at attribute level in order to get majority margins, i.e. a bipolar-valued similarity

graph. Good maximal cliques in this graph, with respect to a fitness measure, are chosen as cluster cores and then expanded to form a complete partition. As the enumeration of all the maximal cliques is well known to be potentially exponential [23], we develop a special meta-heuristic for dealing with the first step. The aim of our method is to achieve a partition that will minimize the differences between the original similarity relation and the relation that is implied by the clustering result.

2 Dual Similarity-Dissimilarity Modelling

To illustrate the relational concepts of similarity and dissimilarity we first present a small didactic problem.

Let us consider in Figure 1 a set of objects $\{a, b, c, d\}$ that are described by four attributes, one cardinal and three ordinal. We may notice that objects a , b and c are quite small, while d is significantly larger. On the second attribute a and b , as well as c and d have the same texture. On the color attribute we notice some objects are dark, and some are light or we could consider each color level to be different. This can be perceived differently by anyone who looks at these objects. On the last attribute, we have the shapes of each object, and we could consider that object a is different from b but similar to the rest, object b is similar to both a and c but different from d and c is also different from d . None of these objects are similar on all attributes, therefore we could consider two objects to be similar overall if they have similar evaluations on a majority of attributes. For example objects a and b have close evaluations on three out of four attributes, therefore they are considered to be globally *similar*. Objects c and d have also three attributes out of four on which they are similar. But on the first attribute, they show a very large difference in evaluations (4 cm compared to 20 cm). Here, we would rather like to say that *we are not sure* if they are similar or not.

Attributes	a	b	c	d
Size	2 cm	3 cm	4 cm	20 cm
Texture	Smooth	Smooth	Rough	Rough
Color	Black	Black	Gray	White
Shape	Square	Circle	Rounded Square	Rectangle

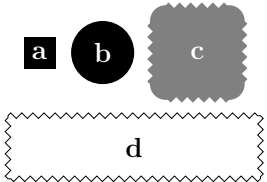


Fig. 1. Objects' evaluations on the four attributes (left) and their schematic representation (right, smooth (resp. zigzagged) lines representing the smooth (resp. rough) texture)

2.1 Pairwise Similarity and Dissimilarity Statements

Let $X = \{x, y, z, \dots\}$ denote a set of n objects. Each object $x \in X$ is described on a set $I = \{i, j, k, \dots\}$ of m attributes of nominal, ordinal and/or cardinal type,

where the actual evaluation x_i may be encoded without loss of generality in the real interval $[m_i, M_i]$ ($m_i < M_i \in \mathbb{R}$). The attributes may not all be of the same significance for assessing the global similarity between the objects. Therefore we assign to the attributes normalized weights $w_i \in [0, 1]$ s.t. $\sum_{i \in I} w_i = 1$, which can be given by the human agent and depend on his knowledge of the problem and his perception of the importance of each attribute in the comparison of the objects.

In order to characterize the *marginal pairwise similarity* and *marginal pairwise dissimilarity* relations between two alternatives x and y of X for each attribute i of I , we use the functions $s_i, d_i : X \times X \rightarrow \{-1, 0, 1\}$ defined as follows:

$$s_i(x, y) := \begin{cases} +1 & , \text{ if } |x_i - y_i| \leq \sigma_i; \\ -1 & , \text{ if } |x_i - y_i| \geq \delta_i; \\ 0 & , \text{ otherwise.} \end{cases} \quad d_i(x, y) := \begin{cases} -1 & , \text{ if } |x_i - y_i| \leq \sigma_i; \\ +1 & , \text{ if } |x_i - y_i| \geq \delta_i; \\ 0 & , \text{ otherwise.} \end{cases} \quad (1)$$

where $0 \leq \sigma_i < \delta_i \leq M_i - m_i, \forall i \in I$ denote marginal similarity and dissimilarity discrimination thresholds. These thresholds are parameters which can be fixed by the human agent according to his a priori knowledge on the data and may be constant and/or proportional to the values taken by the objects being compared. If $s_i(x, y) = +1$ (resp. $d_i(x, y) = +1$) we conclude that x and y are similar (resp. dissimilar) on attribute i . If $s_i(x, y) = -1$ (resp. $d_i(x, y) = -1$) we conclude that x and y are not similar (resp. not dissimilar) on attribute i . When $s_i(x, y) = 0$ (resp. $d_i(x, y) = 0$) we are in doubt whether x and y are, on attribute i , to be considered similar or not similar (resp. dissimilar or not dissimilar). Missing values are also handled by giving an indeterminate $s_i(x, y) = 0$, as we cannot state anything regarding this comparison.

The *weighted similarity* and *weighted dissimilarity* relations between x and y , aggregating all marginal similarity statements and all dissimilarity statements are characterized via the functions $ws, wd : X \times X \rightarrow [-1, 1]$ defined as follows:

$$ws(x, y) := \sum_{i \in I} w_i \cdot s_i(x, y) \quad wd(x, y) := \sum_{i \in I} w_i \cdot d_i(x, y) \quad (2)$$

Again, if $0 < ws(x, y) \leq 1$ (resp. $0 < wd(x, y) \leq 1$) we may assume that it is more sure than not that x is similar (resp. dissimilar) to y ; if $-1 \leq ws(x, y) < 0$ ($-1 \leq wd(x, y) < 0$) we may assume that it is more sure that x is not similar (not dissimilar) to y than the opposite; if, however, $ws(x, y) = 0$ (resp. $wd(x, y) = 0$) we are in doubt whether object x is similar (resp. dissimilar) to object y or not.

Property: The weighted dissimilarity is the *negation* of the weighted similarity relation: $wd = -ws$.

2.2 Taking into Account Strong Dissimilarities

In some cases two objects may be similar on most of the attributes but show a very strong dissimilarity on some other attribute. In this case the objects

cannot be considered overall similar or dissimilar. To model this *indeterminate* situation, we define a *marginal strong dissimilarity* relation between objects x and y with the help of function $sd_i : X \times X \rightarrow \{0, 1\}$ as follows:

$$sd_i(x, y) := \begin{cases} 1 & , \text{ if } |x_i - y_i| \geq \delta_i^+; \\ 0 & , \text{ otherwise.} \end{cases} \quad (3)$$

where δ_i^+ is such that $\delta_i < \delta_i^+ \leq M_i - m_i$ and represents a strong dissimilarity threshold. Again, this threshold is given by the human agent, in accordance with his experience concerning the underlying problem. If $sd_i(x, y) = 1$ (resp. $sd_i(x, y) = 0$) we conclude that x and y are *strongly dissimilar* (resp. not strongly dissimilar) on attribute i .

We consider that two objects x and y of X , described on a set I of attributes, are *overall similar*, denoted $(x \text{ S } y)$, if

1. a weighted majority of the attributes in I validates a similarity situation between x and y and,
2. there is no marginal strong dissimilarity situation observed between x and y .

We formally characterize the *overall similarity* and *overall dissimilarity* relations by functions $s, d : X \times X \rightarrow [-1, 1]$ as follows:

$$s(x, y) := \bigcircledvee (ws(x, y), -sd_1(x, y), \dots, -sd_m(x, y)) \quad (4)$$

$$d(x, y) := \bigcircledvee (wd(x, y), sd_1(x, y), \dots, sd_m(x, y)) \quad (5)$$

where, for $q \in \mathbb{N}_0$, the epistemic disjunction operator $\bigcircledvee : [-1, 1]^q \rightarrow [-1, 1]$ is defined as follows:

$$\bigcircledvee (p_1, p_2, \dots, p_q) := \begin{cases} \max(p_1, p_2, \dots, p_q) & , \text{ if } p_i \geq 0, \forall i \in \{1 \dots q\}; \\ \min(p_1, p_2, \dots, p_q) & , \text{ if } p_i \leq 0, \forall i \in \{1 \dots q\}; \\ 0 & , \text{ otherwise.} \end{cases} \quad (6)$$

For two given alternatives x and y of X , if $ws(x, y) > 0$ and no marginal strong dissimilarity has been detected, $s(x, y) = ws(x, y)$ and both alternatives are considered as overall similar. If $ws(x, y) > 0$ and a strong dissimilarity is detected we do not state that x and y are overall similar or not, and $s(x, y) = 0$. If $ws(x, y) < 0$ and, a strong dissimilarity is observed, then x and y are certainly not overall similar and $s(x, y) = -1$. Finally, if $ws(x, y) = 0$ is observed conjointly with a strong dissimilarity, we will conclude that x and y are indeed not overall similar and $s(x, y)$ is put to -1 .

Property: The overall dissimilarity represents the *negation* of the overall similarity: $d = -s$.

Following this property, we can now say that two objects which are not similar according to this characterization can be called dissimilar.

2.3 The Condorcet Similarity Graph

We call a *Condorcet similarity graph*, denoted $G(X, s^*)$, the three-valued graph associated with the bipolar-valued similarity relation s , where X denotes the set of nodes and function $s^* : X \times X \rightarrow \{-1, 0, 1\}$, named *crisp similarity relation*, weights its set of edges as follows:

$$s^*(x, y) := \begin{cases} +1 & , \text{ if } s(x, y) > 0; \\ -1 & , \text{ if } s(x, y) < 0; \\ 0 & , \text{ otherwise.} \end{cases} \quad (7)$$

Figure 2 presents on the left the encoding of the attributes on real scales and the corresponding thresholds for the example defined at the beginning of this section. On the right we have the bipolar-valued overall similarity relation s . Notice that a, b and c are more similar than not to each other, whereas d is surely dissimilar both from a and b . Besides, d and c appear to be neither similar nor dissimilar. The corresponding Condorcet similarity graph is shown below. Edges valued by -1 are not represented and the zero-valued one is dashed. As a Condorcet similarity graph is always reflexive, we do not represent the loops on the nodes.

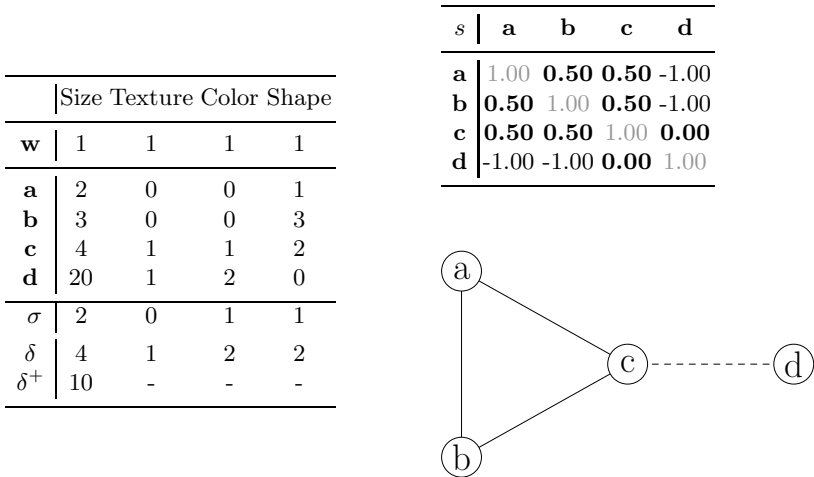


Fig. 2. Encoding of the attributes (left) and bipolar-valued similarity relation with associated Condorcet similarity graph (right)

3 Definition of the Clusters

Ideally, a cluster would have all the objects inside it similar to each other and dissimilar from the rest. In graph theory this may be modeled by a maximal clique, however, we would also need the maximal clique to be totally disconnected from the rest of the graph, which on real data will very rarely be the case. Also there may generally exist a very large number of such maximal cliques, many

overlapping one with the other. Moon and Moser have shown that, in the worst case, the number of maximal cliques in a graph can be exponential [23].

Therefore, in a first stage, we propose to select in the Condorcet similarity graph $G(X, s^*)$ the *best* set of maximal cliques on the +1 arcs (thus containing objects that are, on a majority, similar to each other), which may be considered as cluster cores. In a second stage, we expand these cores into clusters by adding objects that are well connected to them in such a way that we try to maximize the number similarity arcs inside each cluster, and minimize the number of similarity arcs between the clusters.

Let us introduce several fitness measures we will need in the algorithmic approach. Given a Condorcet similarity graph $G(X, s^*)$ and a set $K \subseteq X$ of objects, we define, for each x of X the crisp similarity majority margin $smm^* : X \times \mathcal{P}(X) \rightarrow [-n, n]$ towards the set K :

$$smm^*(x, K) := \sum_{y \in K} s^*(x, y). \quad (8)$$

A large positive value of $smm^*(x, K)$ would show that x is similar to the set K in a consistent manner. A large negative value would mean that x is mostly dissimilar from K .

We define the profile of a set K by the set of all similarity majority margins for $x \in X$.

We will consider a cluster to have a strong profile if it contains strongly positive and/or negative similarity majority margins and reflect this using the core fitness function $f_C^* : \mathcal{P}(X) \rightarrow [-n^2, n^2]$ defined as:

$$f_C^*(K) := \sum_{x \in X} |smm^*(x, K)|. \quad (9)$$

In Figures 3 and 4 we show how two possible cluster cores could be characterised. The examples show a set of 10 objects $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and a possible cluster core $\{1, 2, 3, 4, 5\}$. On the left we have the crisp similarity relation between the core and all the objects in the example and its similarity majority margins. On the right we show a representation of the Condorcet Similarity Graph.

In the first image $\{1, 2, 3, 4, 5\}$ is well connected to objects 6 and 7, and very well disconnected from the rest of the objects. Objects 6 and 7 are connected to 4 out of 5 objects in the maximal clique and can be added later to eventually form a cluster. This can be regarded as a good cluster core. In the second image the same maximal clique is not consistently connected to all the objects outside. Each of them is connected to either 2 or 3 objects inside the core, and this is reflected in the similarity majority margins which take low positive and negative values. This is regarded as a weaker cluster core.

The core fitness we have defined before tells us how well a maximal clique will serve as a cluster core. If for a given maximal clique we cannot find a better one in its vicinity then we can use this one as a core and expand it with objects that are well connected to it. Therefore, in order to achieve a partitioning of the

s^*	1	2	3	4	5	6	7	8	9	10
1	+1	+1	+1	+1	+1	+1	+1	-1	-1	-1
2	+1	+1	+1	+1	+1	+1	-1	-1	-1	-1
3	+1	+1	+1	+1	+1	-1	+1	-1	-1	-1
4	+1	+1	+1	+1	+1	+1	+1	-1	-1	-1
5	+1	+1	+1	+1	+1	+1	+1	-1	-1	-1
smm*	+5	+5	+5	+5	+5	+3	+3	-5	-5	-5

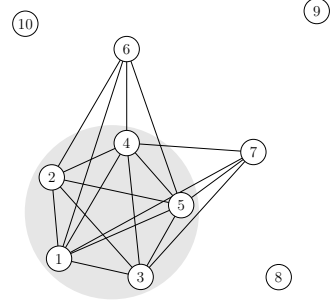


Fig. 3. Well-defined cluster core $K = \{1, 2, 3, 4, 5\}$

s^*	1	2	3	4	5	6	7	8	9	10
1	+1	+1	+1	+1	+1	-1	-1	-1	-1	-1
2	+1	+1	+1	+1	+1	-1	-1	-1	+1	+1
3	+1	+1	+1	+1	+1	-1	+1	+1	-1	-1
4	+1	+1	+1	+1	+1	+1	+1	-1	+1	+1
5	+1	+1	+1	+1	+1	+1	+1	+1	+1	-1
smm*	+5	+5	+5	+5	+5	-1	+1	-1	+1	-1

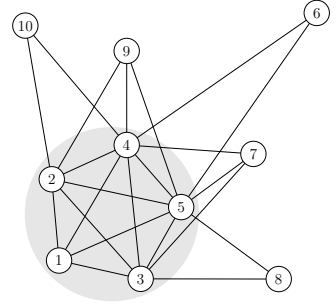


Fig. 4. Less well-defined cluster core $K = \{1, 2, 3, 4, 5\}$

entire dataset we will detect maximal cliques that correspond to local maxima of the core fitness measure. To find these local maxima, we define the neighborhood of a maximal clique as all the maximal cliques that contain at least one object from it.

Let us define now the fitness an alternative x would have as part of a cluster K through function $f^* : X \times \mathcal{P}(X) \rightarrow [-n^2, n^2]$ as:

$$f^*(x, K) := \sum_{y \in X} s^*(x, y) \cdot \text{smm}^*(x, K). \quad (10)$$

If x is mostly similar to K and compares to the rest of the objects in X mostly the same as the objects in K then $f^*(x, K)$ will be high.

Finally we define the fitness of a partition, with respect to the crisp similarity relation, as the outcome of the clustering method through $f_P^* : \mathcal{O}(X) \rightarrow [-n^2, n^2]$, where \mathcal{O} is the set of all possible partitions of X :

$$f_P^*(\mathcal{K}) := \sum_{K \in \mathcal{K}} \sum_{x, y \in K} s^*(x, y) + \sum_{K_1 \neq K_2 \in \mathcal{K}} \sum_{x \in K_1, y \in K_2} -s^*(x, y). \quad (11)$$

As the clustering result will be a partition, we wish to maximize this fitness function and therefore will use it as the criterion to be optimized.

4 Clustering Algorithm

The exact algorithmic approach to find the best partition would be to enumerate all of them and select the one that maximizes the fitness function defined above. However, this approach is not feasible even for small problems. The number of partitions for a problem of size n is given by the Bell number for which an upper bound of $\left(\frac{0.792 \cdot n}{\ln(n+1)}\right)^n$ has been recently given in [4].

Therefore we structure our algorithmic approach in the following four steps:

1. Elicit the thresholds and weights on each attribute from a human agent.
2. Construct the bipolar-valued similarity relation and its associated Condorcet similarity graph according to this preferential information.
3. Find the cluster cores.
4. Expand the cores and achieve a complete partition.

The *first step* is not fully covered in this paper, however this parameter elicitation step is crucial and is one of the distinctive features of our approach. We are currently exploring this step thoroughly. It is inspired from preference elicitation techniques used in multiple criteria decision aid to obtain the preferences of the decision maker (see [21] for a large review of such methods). It should be based both on a direct induction of the parameters and an indirect one, trying to exploit holistic judgements of the human actor on the similarity or dissimilarity of some objects well-known to him.

In the direct elicitation process, the human actor is asked to assign numerical values to the various discrimination thresholds and the importance weights of the attributes, according to his expertise or his perception of the underlying data. Via an indirect process, this information can at any time be complemented by overall judgements of the human actor on some objects. They can be, among others, of the following forms :

- I consider that objects a and b are overall similar (or dissimilar);
- I consider that objects a and b are more similar (or dissimilar) than c and d are.

These judgements are then included as linear constraints on the overall similarity characteristic functions in a linear program whose goal is to determine a set of discrimination thresholds and importance weights in accordance with these inputs (see [22] for a similar approach in multiple criteria decision aid). Note again that this step should be carried over only a small sample of the original dataset.

The *second step* is straight-forward and derives from the definitions in the above section.

In the *third step*, we may use two resolution strategies: exact enumeration of all the maximal cliques and selection of the fittest ones as potential cluster cores, or a population-based metaheuristic approach.

For the exact approach we use the Bron-Kerbosch algorithm [8], with the pivot point improvement from Koch [18]. We then evaluate the fitness of each

maximal clique and compute the neighbourhood matrix from which we retrieve the maximal cliques that are the local maxima of the fitness function. As previously mentioned, the number of maximal cliques in a graph can be exponential [23], making the use of exact approaches for large or even medium clustering problems rapidly intractable.

To overcome this operational problem, we use a population-based meta-heuristic close in structure to evolutionary strategies [28]. As such, the meta-heuristic contains 4 steps: initialization, selection, reproduction and replacement. Each individual in the population is a maximal clique in the Condorcet similarity graph. Our aim is to discover all maximal cliques that are local maxima of our fitness measure.

In the *initialization step* we, first, iteratively generate maximal cliques that do not overlap with each other. After each object has been covered by at least one maximal clique, the rest of the population is then generated randomly.

The *selection step* has a large number of potential variations. We have opted after several tests for the rank-based roulette wheel method.

The *reproduction step* is based on a mutation operator specifically designed for maximal cliques. The maximal clique that will generate a new individual in the population is incrementally stripped with a given probability of its objects and then grown by adding other objects until the property of maximality is reached. The generated population is of equal size with the old one.

In the *replacement step*, all maximal cliques in the current population that are local maxima of the fitness measure, based however on the limited exploration of their neighborhoods that has been done at previous iterations, are kept in the new population. The rest of the individuals to be kept are selected at random, in order to maintain a good exploration of the search space.

The *last step* orders all the objects that were not included in a core based on their best fitness to be added to a core. The majority margins heuristic, in fact, tells us how many relations are in accordance with the decision to add the object to a particular core, therefore iteratively taking the best pair of object and core and adding that object to the core is well justified considering our goals to extract a partition that is in most accordance to the original similarity relation.

5 Results

We would like to present some results on the 2010-11 Times Higher Education World University Ranking data [1]. The dataset contains 199 universities evaluated on 13 separate indicators designed to capture a broad range of activities, from teaching and research to knowledge transfer. These elements are brought together into five attributes on scales from 0 to 100: Teaching environment (T), International mix (I), Industry income (Ind), Research (R) and Citations (C).

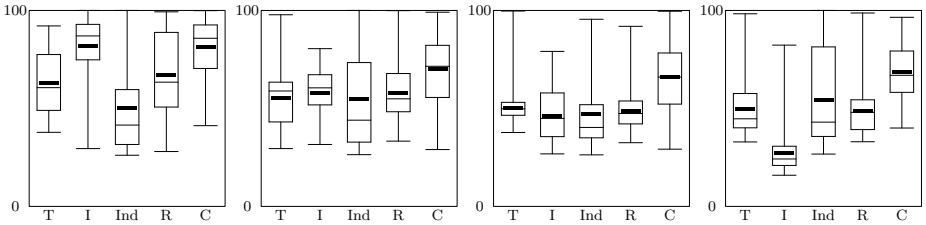
We do not wish to rank the universities, as it could be easily misunderstood, but to find clusters of similar universities according to two persons' viewpoints. Let's consider person A to be a student who is more interested in the teaching environment and international side of the universities, and let person B be a

Table 1. Weights extracted to reflect two persons' viewpoints)

Person	T	I	Ind	R	C
A	0.31	0.42	0.10	0.10	0.05
B	0.05	0.10	0.10	0.42	0.31

postdoc who is more interested in looking at the research environment characterising each university. For each person we will find the set of clusters which make most sense according to their viewpoints.

Due to the fact that each attribute is constructed and brought to the same scale, we take as thresholds $\sigma = 5\%$, $\delta = 10\%$, $\delta^+ = 50\%$ of the scales range for each attribute. We then select a set of weights in accordance with each person's point of view as seen in Table 1.

**Fig. 5.** Person A cluster box plots

For person A we find 4 clusters for which we present the boxplots of the objects inside them in Figure 5. We notice how we have grouped together universities with close evaluations on the second attribute which was deemed as most important for person A. In the first cluster we find universities with high evaluations on the International attribute (I), medium-high values in the second, medium-low in the third, and very low in the last. The Teaching attribute (T) also differentiates well the clusters, with progressively lower evaluations from the first to third clusters, while slightly higher on the last.

In the case of the second point of view we find 6 clusters where universities are grouped together mostly based on the fourth attribute, Research (R), as seen in Figure 6. Also the last attribute, Citations (C), is well defining for each cluster.

As we wish that the clustering result would contain clusters of objects that are all similar to each other and dissimilar from the rest, we could model this by a similarity relation that contains +1 values between objects inside the same cluster and -1 values between objects in different clusters. To measure the quality of the results we use the fitness of a partition that we defined in Section 3. We compute this fitness according to the similarity relation defined by each point of view that we have modeled. We also compare our results to some well-known algorithms like K-means [20], Single-Link and Complete-Link Agglomerative

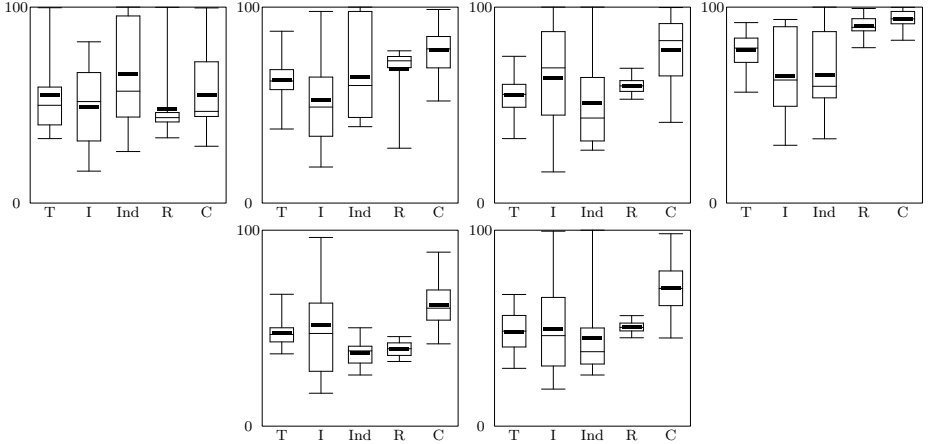


Fig. 6. Person B cluster box plots

Table 2. Fitness of clusterings according to two persons' viewpoints

	Person A POV	Person B POV
GAMMA-S for A	0.784	0.707
GAMMA-S for B	0.763	0.853
K-Means	0.710	0.698
SL AHC	0.204	0.213
CL AHC	0.682	0.728
PAM	0.704	0.791

Hierarchical Clustering [15] and Partitioning Around Medoids [17]. For all of these, we give them as input the number of clusters we have found with our approach.

We further present some results on a few well-known datasets such as the Iris, Wines and Breast Cancer datasets from the UCI Machine Learning Repository [12]. Each of these datasets comes, based on the analysis of some experts on the corresponding problem, with the set of clusters we should obtain. Therefore a good criterion for evaluation is the *Jaccard Coefficient* [14] in order to measure how close the results of the clustering algorithms are to the desired clusters.

Due to the absence of the experts that proposed the clusters for each problem we have used $\sigma = 10\%$, $\delta = 20\%$ and $\delta^+ = 70\%$ of the value range on each attribute as thresholds and given equal significance to all attributes. We show, however, on the Iris dataset which contains 150 objects defined on 4 attributes, that if such an interaction were possible, we could have extracted a threshold set that would bring the clustering results of GAMMA-S very close to the clusters that were proposed by the experts. Therefore, with $\sigma_1 = 0\%$, $\sigma_2 = 33\%$, $\sigma_3 = 25\%$, $\sigma_4 = 12\%$ of the value ranges of each attribute and the dissimilarity

Table 3. Average results on Jaccard Coefficient (standard deviations in brackets)

Algorithm/Dataset	Iris	Wines	Breast Cancer
K-means	0.529 (0.118)	0.461 (0.100)	0.554 (0.130)
SL AHC	0.589 (-)	0.336 (-)	0.531 (-)
CL AHC	0.622 (-)	0.805 (-)	0.588 (-)
PAM	0.712 (0.007)	0.734 (0.000)	0.622 (0.025)
GAMMA-S	0.525 (-)	0.766 (-)	0.539 (0.028)

thresholds higher by 1% the Jaccard Coefficient of the GAMMA-S result would be equal to 0.878.

Except for our algorithm, we have given the a priori knowledge regarding how many clusters the outcome should have to the rest. The results come from running every non-deterministic algorithm 100 times over each instance. For the first two datasets, due to their small size, we have used the exact approach of our algorithm.

We notice overall that *GAMMA-S* performs very well considering the assumptions we make. We also notice that if we would be able to extract preferential information from a person who wants to cluster this data, we would get results more in accordance with his point of view on the problem. In addition we neither need to provide commensurable cardinal attributes nor an a priori number of clusters.

6 Conclusions and Perspectives

We conclude that our clustering method does indeed give consistent results, however without any requirements on the data, as all kinds of attribute types can be considered. Furthermore, imprecision, uncertainties and even missing values can easily be handled by the similarity relation defined in this article. There are many improvements that could be done to increase the performance of our approach, which will be explored in the future. At the moment we need to explore elicitation techniques for the parameters of the model. We also wish to present more extensive results on datasets on which we could have this interaction with a real person. The complexity of the algorithm should be improved by fine-tuning the meta-heuristic, while the final result our method proposes could be further improved by means of a local search method.

References

1. The Times Higher Education World University Rankings (2010-2011)
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data (2005)
3. Ankerst, M., Breunig, M.M., Kriegel, H., Sander, J.: Optics: ordering points to identify the clustering structure. In: International Conference on Management of Data, pp. 49–60 (1999)

4. Berend, D., Tassa, T.: Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics* 30, 185–205 (2010)
5. Bisdorff, R.: Logical foundation of fuzzy preferential systems with application to the electre decision aid methods. *Computers & Operations Research* 27, 673–687 (2000)
6. Bisdorff, R.: Electre-like clustering from a pairwise fuzzy proximity index. *European Journal of Operational Research* 138(2), 320–331 (2002)
7. Bisdorff, R.: On clustering the criteria in an outranking based decision aid approach. In: *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pp. 409–418. Springer, Heidelberg (2008)
8. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* 16(9), 575–577 (1973)
9. Cheeseman, P., Stutz, J.: *Bayesian Classification (AutoClass): Theory and Results*, ch. 6, pp. 62–83. AAAI Press, MIT Press (1996)
10. Du, N., Wu, B., Pei, X., Wang, B., Xu, L.: Community detection in large-scale social networks. In: *WebKDD/SNA-KDD 2007: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pp. 16–25. ACM (2007)
11. Fortunato, S.: Community detection in graphs. *Physics Reports* 486(3-5), 75–174 (2010)
12. Frank, A., Asuncion, A.: *UCI machine learning repository* (2010)
13. Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. In: Haas, L., Drew, P., Tiwary, A., Franklin, M. (eds.) *SIGMOD 1998: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 73–84. ACM Press (1998)
14. Jaccard, P.: Nouvelles recherches sur la distribution florale. *Bulletin de la Societe Vaudoise de Sciences Naturelles* 44, 223–370 (1908)
15. Jain, A., Dubes, R.: *Algorithms for clustering data*. Prentice-Hall, Inc. (1988)
16. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *ACM Computing Survey* 31(3), 264–323 (1999)
17. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data An Introduction to Cluster Analysis*. Wiley Interscience (1990)
18. Koch, I.: Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science* 250(1-2), 1–30 (2001)
19. Kohonen, T.: *Self-organising maps*. Information Sciences (1995)
20. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 (1967)
21. McLachlan, G., Krishnan, T.: *The EM algorithm and extensions*. Wiley Series in Probability and Statistics, 2nd edn. Wiley (2008)
22. Meyer, P., Marichal, J.-L., Bisdorff, R.: Disaggregation of bipolar-valued outranking relations. In: An, L.T.H., Bouvry, P., Tao, P.D. (eds.) *MCO. CCIS*, vol. 14, pp. 204–213. Springer, Heidelberg (2008)
23. Moon, J., Moser, L.: On cliques in graphs. *Israel Journal of Mathematics* 3(1), 23–28 (1965)
24. Mousseau, V.: Elicitation des préférences pour l’aide multicritère à la décision (2003)
25. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69(2) (2004)

26. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043), 814–818 (2005)
27. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: *Proceedings of 24rd International Conference on Very Large Data Bases VLDB 1998*, pp. 428–439. Morgan Kaufmann (1998)
28. Talbi, E.: *Metaheuristics - From Design to Implementation*. Wiley (2009)
29. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: *SIGMOD 1996: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pp. 103–114. ACM Press (1996)
30. Zhou, B., Cheung, D.W., Kao, B.: A fast algorithm for density-based clustering in large database. In: Zhong, N., Zhou, L. (eds.) *PAKDD 1999*. LNCS (LNAI), vol. 1574, pp. 338–349. Springer, Heidelberg (1999)

A False Negative Maximal Frequent Itemset Mining Algorithm over Stream*

Haifeng Li and Ning Zhang

School of Information, Central University of Finance and Economics
Beijing, China, 100081

mydlhf@126.com, zhangning75@sina.com

Abstract. Maximal frequent itemsets are one of several condensed representations of frequent itemsets, which store most of the information contained in frequent itemsets using less space, thus being more suitable for stream mining. This paper focuses on mining maximal frequent itemsets approximately over a stream landmark model. We separate the continuously arriving transactions into sections and maintain them with 3-tuple lists indexed by an extended direct update tree; thus, an efficient algorithm named *FNMFIMoDS* is proposed. In our algorithm, we employ the Chernoff Bound to perform the maximal frequent itemset mining in a false negative manner; plus, we classify the itemsets into categories and prune some redundant itemsets, which can further reduce the memory cost, as well guarantee our algorithm conducting with an incremental fashion. Our experimental results on two synthetic datasets and two real world datasets show that with a high precision, *FNMFIMoDS* achieves a faster speed and a much reduced memory cost in comparison with the state-of-the-art algorithm.

1 Introduction

Frequent itemset mining is a traditional and important problem in data mining. An itemset is frequent if its support is not less than a threshold specified by users. Traditional frequent itemset mining approaches have mainly considered the problem of mining static transaction databases. In these methods, transactions are stored in secondary storage so that multiple scans over data can be performed. Three kinds of frequent itemset mining approaches over static databases have been proposed: reading-based [3], writing-based [15], and pointer-based [18]. [14] presented a comprehensive survey of frequent itemset mining and discussed research directions.

Many methods focusing on frequent itemset mining over a stream have been proposed. [13] proposed *FP-Stream* to mine frequent itemsets, which was efficient when the average transaction length was small; [22] used lossy counting to mine

* This research is supported by the National Science Foundation of China(61100112), and Program for Innovation Research in Central University of Finance and Economics.

frequent itemsets; [7], [8], and [9] focused on mining the recent itemsets, which used a regression parameter to adjust and reflect the importance of recent transactions; [27] presented the *FTP-DS* method to compress each frequent itemset; [10] and [1] separately focused on multiple-level frequent itemset mining and semi-structure stream mining; [12] proposed a group testing technique, and [17] proposed a hash technique to improve frequent itemset mining; [16] proposed an in-core mining algorithm to speed up the runtime when distinct items are huge or minimum support is low; [19] presented two methods separately based on the average time stamps and frequency-changing points of patterns to estimate the approximate supports of frequent itemsets; [5] focused on mining a stream over a flexible sliding window; [20] was a block-based stream mining algorithm with *DSTree* structure; [23] used a verification technique to mine frequent itemsets over a stream when the sliding window is large; [11] reviewed the main techniques of frequent itemset mining algorithms over data streams and classified them into categories to be separately addressed. Given these algorithms, the runtime could be reduced, but the mining results were huge when the minimum support was low; consequently, the condensed representations of frequent itemsets including closed itemsets [25], maximal itemsets [31], free itemsets [4], approximate k-sets [2], weighted itemsets [28], and non-derivable itemsets [6] were proposed; in addition, [26] focused on discovering a minimal set of unexpected itemsets.

The concept of maximal frequent itemsets (*MFI*) was first proposed in 1998, an itemset is maximal frequent itemset if its support is frequent and it is not covered by other frequent itemsets. We will discuss the details in Section 2. Maximal frequent itemsets are one of the condensed representations, which only store the non-redundant cover of frequent itemsets, resulting in a space cost reduction.

Many maximal frequent itemset mining algorithms were proposed to improve the performance. The main considerations focused on developing new data retrieving method, new data pruning strategy and new data structure. Yang used directed graphs in [35] to obtain maximal frequent itemsets and proved that maximal frequent itemset mining is a $\#p$ problem. The basic maximal frequent itemset mining method is based on the *a priori* property of the itemset. The implementations were separated into two types: One type is an improvement of the *a priori* mining method, a breadth first search [32], with utilizing data pruning, nevertheless, the candidate results are huge when an itemset is large; a further optimization was the bottom-up method, which counted the weight from the largest itemset to avoid superset checking, also, the efficiency was low when the threshold was small. Another one used depth first search [33] to prune most of the redundant candidate results, which, generally, is better than the first type. In these algorithms, many optimized strategies were proposed [34] [31]: The candidate group has a head itemset and a tail itemset, which can quickly build different candidate itemsets; the super-itemset pruning could immediately locate the right frequent itemset; the global itemset pruning deleted all the subitemsets according to the sorted itemsets; the item dynamic sort strategy built heuristic rules to directly obtain the itemsets with high support, which was extended by

a further pruning based on tail itemset; the local check strategy got the related maximal frequent itemsets with the current itemset.

Many stream mining algorithms for maximal frequent itemsets were proposed to improve the performance. [21] proposed an increasing algorithm *estDec+* based on *CP-tree* structure, which compressed several itemsets into one node according to their frequencies; thus, the memory cost can be flexibly handled by merging or splitting nodes. Furthermore, employing an *isFI-forest* data structure to maintain itemsets, [30] presented *DSM-MFI* algorithm to mine maximal frequent itemsets. Moreover, considering maximal frequent itemset is one of the condensed representation, [24] proposed *INSTANT* algorithm, which stored itemsets with frequencies under a specified minimum support, and compared them to the new transactions to obtain new itemsets; Plus, [29] presented an improved method *estMax*, which predicted the type of itemsets with their defined maximal life circle, resulting in advanced pruning.

In this paper, we address this problem and propose a Chernoff Bound based method named *FNMFIMoDS* (**F**alse **N**egative **M**aximal **F**requent **I**temset **M**ining **o**ver **D**ata **S**tream) on a stream landmark model.

The rest of this paper is organized as follows: In *Section 2* we define the mining problem with presenting the preliminaries of frequent itemsets, maximal frequent itemsets, and Chernoff Bound method. *Section 3* presents a naive method based on Chernoff Bound. *Section 4* illustrates the data structure and our algorithm in detail. *Section 5* evaluates the performance of our algorithm with experimental results. Finally, *Section 6* concludes this paper.

2 Preliminaries

2.1 Frequent Itemsets

Given a set of distinct items $\Gamma = \{i_1, i_2, \dots, i_n\}$ where $|\Gamma| = n$ denotes the size of Γ , a subset $X \subseteq \Gamma$ is called an itemset; suppose $|X| = k$, we call X a k -itemset. A concise expression of itemset $X = \{x_1, x_2, \dots, x_m\}$ is $x_1x_2 \dots x_m$. A database $D = \{T_1, T_2, \dots, T_v\}$ is a collection wherein each transaction is a subset of Γ , namely an itemset. Each transaction T_i ($i = 1 \dots v$) is related to an id, i.e., the id of T_i is i . The absolute support (*AS*) of an itemset α , also called the weight of α , is the number of transactions which cover α , denoted $\Lambda(\alpha, D(v))$, $\Lambda(\alpha)$ in short, is $\{|T| | T \in D \wedge \alpha \subseteq T\}$; the relative support (*RS*) of an itemset α is the ratio of *AS* with respect to $|D|$, denoted $\Lambda_r(\alpha) = \frac{\Lambda(\alpha)}{v}$. Given a relative minimum support λ ($0 \leq \lambda \leq 1$), itemset α is frequent if $\Lambda_r(\alpha) \geq \lambda$.

2.2 Maximal Frequent Itemsets

A maximal itemset is a largest itemset in a database D , that is, it is not covered by other itemsets. A maximal frequent itemset is both maximal and frequent in D .

Definition 1. Given an relative support λ , an itemset α is maximal frequent itemset if it is frequent and it is not covered by other frequent itemsets, denoted $\Lambda_r(\alpha) \geq \lambda \wedge \nexists \beta (\beta \supset \alpha \wedge \Lambda_r(\beta) \geq \lambda)$.

2.3 Chernoff Bound

Given n independent value $o_1, o_2, \dots, o_n, o_{n+1}, \dots$ according with Bernoulli trial and probable value p , $Pr[o_i = 1] = p$ and $Pr[o_i = 0] = 1 - p$ is satisfied. Let r be the actual value of $Pr[o_i = 1]$, then the expected value of r is np , $\forall \eta > 0$, $Pr\{|r - np| \geq np\eta\} \leq 2e^{-\frac{np\eta^2}{2}}$. Let $\bar{r} = \frac{r}{n}$, $Pr\{|\bar{r} - p| \geq p\eta\} \leq 2e^{-\frac{np\eta^2}{2}}$. Let $\varepsilon = p\eta$; thus, $Pr\{|\bar{r} - p| \geq \varepsilon\} \leq 2e^{-\frac{n\varepsilon^2}{2p}}$. Let $\delta = 2e^{-\frac{n\varepsilon^2}{2p}}$, then the average probable value of $o_i = 1$ beyond the range of $[p - \varepsilon, p + \varepsilon]$ with a probability less than δ , in which $\varepsilon = \sqrt{\frac{2p\ln(2/\delta)}{n}}$. That is to say, for a sequence of transactions $D = \{t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_N\}$ where n is the number of first arrived transactions and $n \ll N$. The actual support $A_r(X, D(N))$ for itemset X is within the range of $[A_r(X, D(n)) - \varepsilon, A_r(X, D(n)) + \varepsilon]$ with probability larger than $1 - \delta$.

The *FDPM* algorithm [36] substituted p with the minimum support λ , the rationale is that λ is the minimum support, which is satisfied represents all the other higher supports are also satisfied. For an instance, suppose $\lambda = 0.1$, $\delta = 0.1$, and $\varepsilon = 0.01$, according to Chernoff Bound, $n \approx 5991$, i.e., for 5991 transactions, the actual support of itemset I is between 0.09 and 0.11 with probability larger than 0.9. [37] try to mine all the results with one stream scan: The stream is split into sections, and the frequent itemsets from the previous section will be used as the candidate itemsets of the next section, Chernoff Bound is also used to guarantee the precision. As can be seen, both algorithms aim to obtain the frequent itemsets. We will prove that the Chernoff Bound method can also be effectively used in maximal frequent itemset mining based on our definition.

3 A Naive Method

As can be seen from our addressing problem, we can get a naive method to obtain the maximal frequent itemsets over stream using Chernoff Bound. We can employ *FDPM* method to mine the frequent itemsets, and use a traditional maximal frequent itemset mining method to get the results for each arriving sections. The drawback is, we have to store all the frequent itemsets and potential frequent itemsets of the existing transactions, which may consume much more memory and cannot achieve our destination, i.e., using maximal itemsets to save the memory cost; on the other hand, if we only store the maximal frequent itemsets instead of all frequent itemsets, we cannot guarantee the algorithm performing with an incremental fashion since some itemsets information is missed.

4 *FNMFIMoDS* Algorithm

In this section, we will introduce the data structure and the detailed algorithm based on our assumption, which will be verified by our experiments.

4.1 Data Structure

We employ a 3-tuple $\langle \theta, A_r, \varpi \rangle$ list to store the data synopsis, in which θ represents the itemset, A_r is the relative support, and ϖ refers the itemset category. The itemsets are split into six categories, which will be described in detail after we present our rationale.

Given the minimum support λ , the probability parameter δ , and the dataset D , if an itemset X satisfies $\Lambda(X, D) \geq \lambda + \sqrt{\frac{2\lambda \ln(2/\delta)}{n}}$, we call X the actual frequent itemset(*AF*); if $\lambda + \sqrt{\frac{2\lambda \ln(2/\delta)}{n}} > \Lambda(X, D) \geq \lambda$, then we call X the shifty frequent itemset(*SF*); if $\lambda > \Lambda(X, D) \geq \lambda - \sqrt{\frac{2\lambda \ln(2/\delta)}{n}}$, then we call X the possible frequent itemset(*PF*). Otherwise, X is called the infrequent itemset(*IF*).

For an itemset of one type, it may be covered or not covered by other type itemsets. Consequently, we show all covering possibilities in Fig.1. In this figure, we present the types of the covered itemset at the first column, and present the type of the covering itemset at the first line. For example, if an itemset I is an actual frequent itemset belonging to the type at column 1 and line 2, and it is covered by an actual frequent itemset J belonging to the type at column 2 and line 1, then we define I as a maximal frequent itemset(*MF*); on the contrary, if it is covered by a possible frequent itemset, then we define it as an un-maximal frequent itemset(*UMF*). Plus, if an itemset belonging to the left type cannot be covered by an itemset belonging to the top type, we use \otimes to denote it; an example is, an possible frequent itemset will never be covered by an actual frequent itemset. We classify all itemsets into six categories w.r.t. their current types and future types, which are the surrounding types of the current types in Fig.1.

Definition 2(Actual Un-Maximal Frequent Itemset). If an itemset X is an actual frequent itemset, and it is covered by any other actual frequent itemsets, it is called an actual un-maximal frequent itemset(*AUMF*).

Definition 3(Actual Inter-Maximal Frequent Itemset). If an itemset X is an actual frequent itemset and covered by shifty frequent itemsets, it is called an actual inter-maximal frequent itemset(*AIMF*).

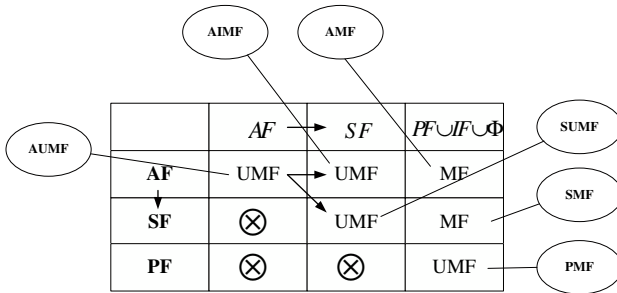


Fig. 1. Covering Relationship

Definition 4(Actual Maximal Frequent Itemset). If an itemset X is an actual maximal frequent itemset, and it is covered by possible frequent itemsets, infrequent itemsets or none itemsets, it is called an actual maximal frequent itemset(AMF).

Definition 5(Shifty Un-Maximal Frequent Itemset). If an itemset X is a shifty frequent itemset and covered by shifty frequent itemsets, it is called a shifty un-maximal frequent itemset($SUMF$).

Definition 6(Shifty Maximal Frequent Itemset). If an itemset X is a shifty frequent itemset, and it is covered by possible frequent itemsets, infrequent itemsets, or none itemsets, it is called a shifty maximal frequent itemset(SMF).

Definition 7(Possible Maximal Frequent Itemset). If an itemset X is a possible frequent itemset, it is a possible maximal frequent itemset(PMF).

4.2 Mining Strategies

We will discuss the properties of our definitions, and present the pruning strategies according to their properties as follows.

Strategy 1. If an itemset I is AMF , it can be output directly as the result, and it may become a shifty frequent itemset, then we will reclassify it into SMF category.

Strategy 2. If an itemset I is $AUMF$, it will not change to maximal frequent itemset in recent future; thus, we can prune them from memory,also, we can ignore processing it.

Strategy 3. If an itemset I is $AIMF$, it may become AMF if the itemsets covering it become possible frequent itemsets, or become $AUMF$ if the itemsets covering it become an actual frequent itemset. Consequently, we have two strategies to maintain actual inter-maximal frequent itemsets. One is used in our paper, that is, we assume that most shifty frequent itemsets will maintain their original type or become actual frequent, then we can prune the actual inter-maximal frequent itemsets directly since they will be covered by at least a frequent itemset; another is based on the contrary assumption, thus we will store and process the actual inter-maximal frequent itemsets.

Strategy 4. If an itemset I is SMF , and it becomes an actual frequent itemset, it is reclassified into AMF category, and all its subsets will be pruned no matter what their original categories are; or, if it becomes a possible frequent itemset, it is reclassified into PMF category, and its subsets may be reclassified from $SUMF$ to SMF .

Strategy 5. If an itemset I is $SUMF$, and it becomes an actual frequent itemset, it is reclassified into $AIMF$ and pruned; or, if it becomes a possible frequent itemset, then it is reclassified into PMF category.

Strategy 6. If an itemset I is PMF , and it becomes a shifty frequent itemset, then it is reclassified into $SUMF$ category or SMF category, if its subsets are AMF , they will be pruned.

4.3 $FNMFIMoDS$

According to our mining strategies, we propose our algorithm $FNMFIMoDS$. As can be seen from Alg. 1, our algorithm can be separated into three parts.

First, we will generate the new itemsets based on the new arriving transactions, with which we update the existed itemsets support. Second, we recompute the new ε_n , prune the new infrequent itemsets, and reclassify each itemset. Finally, we can output the actual maximal frequent itemsets and the shifty maximal frequent itemsets as the results on demand of users. We will discuss it in detail as follows.

Support Updating. When we update the itemsets support, the existed actual un-maximal frequent itemsets and the actual inter-maximal frequent itemsets has been pruned, but the fact is that they may occurs in the new arriving transactions and they may be the new generated frequent itemsets. Consequently, if they are actual un-maximal frequent itemsets, we use the the support of their covering itemset as their supports; if they are actual inter-maximal frequent itemsets, we use $\lambda + \varepsilon_n$ as their supports. Thus, it can be guaranteed that the actual frequent itemsets fall into the adjacent type, i.e., the actual frequent itemsets will become shifty frequent at most.

Indexing Data. To speed up itemsets comparison and results output, we will build the index on the 3-tuples in F_1 . Since our aim is to reduce the memory cost, the index will be as simple as possible. Consequently, we use an extended lexicographical ordered direct update tree ($EDIU$ tree) rather than a traditional prefix tree or an enumeration tree. In our $EDIU$ tree, the root node is the itemset includes all distinct items, and all the descend nodes are the subsets. We link each itemset with their subsets, which are sorted by their lexicographical orders. The advantages of our index are as follows. First, the $EDIU$ tree is simple, only the proper itemsets are stored, the only added information is the pointers between itemsets. Second, when we need to find or compare itemsets, the redundant computing will be ignored. Third, when we prune some itemsets, which is a frequent operation especially in our algorithm, the pruning efficiency is high since most itemsets can be deleted in a cascaded matter.

5 Experimental Results

All experiments were implemented with $C++$, compiled with *Visual Studio 2005* running on *Windows XP* and executed on a *Core2 DUO CPU 2.0GHz* PC with *2GB* RAM.

We used 2 synthetic datasets and 2 real-life datasets, which are well-known benchmarks for frequent itemset mining. The $T10I4D100K$ and $T40I10D100K$

Algorithm 1. FNMFiMoDS Function

Require: F_1 : Existing 3-tuples of frequent itemsets and potential frequent itemsets;
 n_1 : The number of arrived transactions;
 F_2 : New 3-tuples of frequent itemsets and potential frequent itemsets;
 n_2 : The number of new arriving transactions;
 λ : Minimum support;
 φ : Probability;
 ε : False negative parameter;

- 1: $n_2 = \frac{2+2\ln(2/\varphi)}{\lambda}$; [36]
- 2: **for** each n_2 new arriving transactions **do**
- 3: obtain F_2 ;
- 4: $n_1 = n_1 + n_2$;
- 5: $F_1 = F_1 \cup F_2$;
- 6: $\varepsilon = \sqrt{\frac{2s\ln(2/\varphi)}{n_1}}$;
- 7: prune the new infrequent itemsets from F_1 ;
- 8: **for** each 3-tuple I in F_1 **do**
- 9: **if** $I.\varpi = PMF$ and $I.A_r \geq \lambda$ **then**
- 10: $I.\varpi = SMF$ or $SUMF$;
- 11: prune subset J of I from F_1 ;
- 12: **if** $I.\varpi = SUMF$ and $I.A_r \geq \lambda + \varepsilon$ **then**
- 13: prune I from F_1 ;
- 14: **else if** $I.\varpi = SUMF$ and $I.A_r < \lambda$ **then**
- 15: $I.\varpi = PMF$;
- 16: **if** $I.\varpi = SMF$ and $I.A_r \geq \lambda + \varepsilon$ **then**
- 17: $I.\varpi = AMF$;
- 18: prune all subsets of I from F_1 ;
- 19: **else if** $I.\varpi = SMF$ and $I.A_r < \lambda$ **then**
- 20: $I.\varpi = PMF$;
- 21: decide the type of the subset J of I ;
- 22: **if** $I.\varpi = AMF$ and $I.A_r < \lambda + \varepsilon$ **then**
- 23: $I.\varpi = SMF$;
- 24: output 3-tuple I if $I.\varpi = AMF \vee I.\varpi = SMF$ from F_1 on demand;

datasets are generated with the IBM synthetic data generator. The *KOSARAK* dataset contains the click-stream data of a Hungarian online news portal. The *MUSHROOM* dataset contains characteristics from different species of mushrooms. The data characteristics are summarized in Tab. [1](#).

The *estMax* algorithm in [\[29\]](#) is a state-of-the-art method for mining maximal frequent itemsets over stream; thus, we use it as the evaluated method for comparison. In *estMax*, without loss of generality, we configure the fixed parameter $S_{sig} = 0.1$ and $S_{err} = 0.01$; also, we employ our presented naive algorithm as another evaluated method. In our algorithm, we configure the Chernoff Bound parameter as a fixed value, that is, $\delta = 0.1$, which denotes a 10 percent probability for mistaken deleting the actual maximal frequent itemsets.

Table 1. Data Characteristics

DataSet	nr. of trans.	avg. trans. length	min. trans. length	max. trans. length	nr. of items	trans. corr.
T10I4D100K	100 000	10.1	1	29	870	86.1
T40I10D100K	100 000	39.6	4	77	942	23.8
KOSARAK	990 002	7.1	1	2497	36 841	5188.8
MUSHROOM	8 124	23	23	23	119	5.2

5.1 Running Time Cost and Memory Cost Evaluation

As shown in Fig 2, when the minimum support decreases, the running time cost of these three algorithms increase over all datasets. Our algorithm is the best in runtime cost, the reason is that our algorithm prunes some useless computing with classifying the itemsets, as well employs an *EDIU* tree to index itemsets. The naive method is the worst in runtime cost, since we almost use no optimizations for it; nevertheless, the false negative technique reduces the comparing count, thus the running time decreases, and the computing efficiency can reach to that of *estMax* on the *MUSHROOM* dataset and *T40I10D100K* dataset.

As shown in Fig 3, our algorithm reaches to a much lower memory cost than the naive method and *estMax*. That is because our method prune some infrequent itemsets and almost all un-maximal frequent itemsets when the new transactions arrive. Also, the memory cost of the naive method is lower than that of *estMax*, since the naive method is a false negative algorithm, but *estMax* is a false positive algorithm.

5.2 Precision and Recall

Our algorithm, the naive method and *estMax* are all approximate methods, but with employing the Chernoff Bound method, our algorithm and the naive method are much more efficiency in memory cost. On the other hand, some true results may be deleted according to our error analysis. Since the error comparison of frequent itemsets has been conducted in *FDPM* algorithm, here we use precision and recall to present the approximation of the maximal frequent itemsets, which are defined as follows: For an actual collection A and a computed one A' , the precision of A' is $P = \frac{A \cap A'}{A'}$, and the recall of A' is $R = \frac{A \cap A'}{A}$. The larger the precision and recall, the closer between A and A' ; as an example, when $A = A'$, $P = R = 1$.

As shown in Fig 4, the precisions of all the three algorithms are 100% over different datasets, that is because they all can obtain the true maximal frequent itemsets, nevertheless, our algorithm obtain the true results based on a least number of frequent itemsets, which has been discussed in our memory cost experiments. As shown in Fig 5, even though we set a probability of 10 percent to be the wrong results, the experimental results are much better. Since the naive method stores all the un-maximal frequent itemsets, it can obtain all the maximal frequent itemsets; further, the recall of our algorithm is almost the same

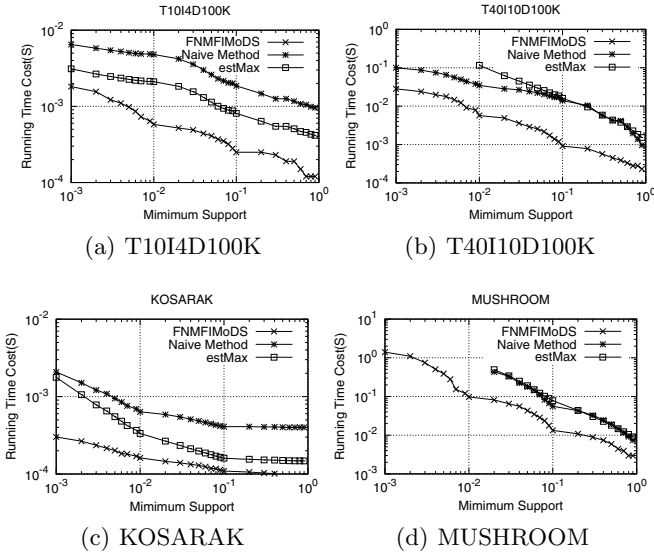


Fig. 2. Running time cost VS. Minimum support

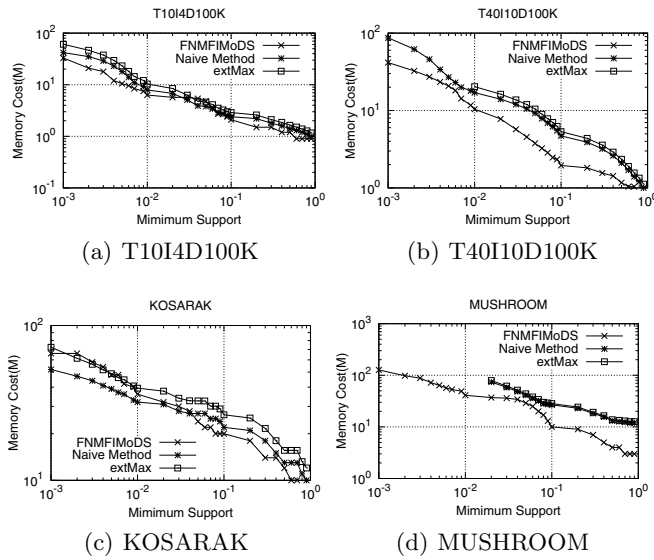


Fig. 3. Memory Cost VS. Minimum support

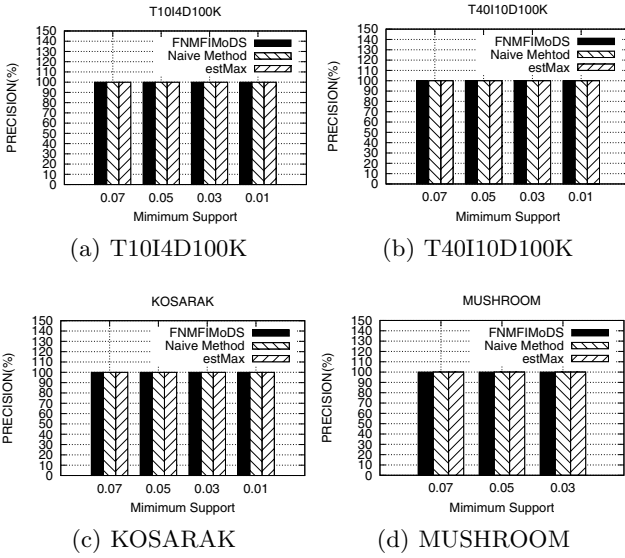


Fig. 4. Precision of maximal frequent itemsets VS. Minimum support

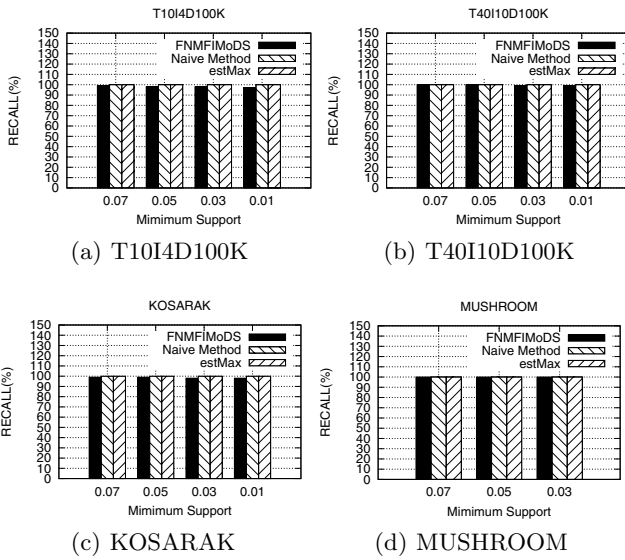


Fig. 5. Recall of maximal frequent itemsets VS. Minimum support

to that of the naive method and *estMax* with a little lower, that is to say, our algorithm mistakenly deletes little real results.

6 Conclusions

In this paper we considered a problem, which is how to mine maximal frequent itemset over stream using a false negative method, and then proposed our method *FNMFIMoDS*. In our algorithm, we used Chernoff Bound to prune the infrequent itemsets; plus, we classified the itemsets into categories to prune the un-maximal frequent itemsets, which still can guarantee that we obtain the proper itemsets; thus, our algorithm was able to perform in an incremental manner. Furthermore, we employed an extended direct update tree to index the itemsets, which can raise the computing efficiency. Our experimental results showed that our algorithm was more efficient in memory cost and running time cost in comparison with the state-of-the-art maximal frequent itemset mining algorithm.

References

1. Asai, T., Arimura, H., Abe, K., Kawasoe, S., Arikawa, S.: Online Algorithms for Mining Semi-structured Data Stream. In: Proc. ICDM (2002)
2. Afrati, F., Gionis, A., Mannila, H.: Approximating a Collection of Frequent Sets. In: Proc. SIGKDD (2004)
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. VLDB (1994)
4. Boulicaut, J., Bykowski, A., Rigotti, C.: Free-sets: a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery* 7, 5–22 (2003)
5. Calders, T., Dexters, N., Goethals, B.: Mining Frequent Itemsets in a Stream. In: Proc. ICDM (2007)
6. Calders, T., Goethals, B.: Mining All Non-Derivable Frequent Itemsets. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, p. 74. Springer, Heidelberg (2002)
7. Chang, J.H., Lee, W.S.: Decaying Obsolete Information in Finding Recent Frequent Itemsets over Data Stream. *IEICE Transaction on Information and Systems* 87(6), 1588–1592 (2004)
8. Chang, J.H., Lee, W.S.: A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams. *Journal of Information Science and Engineering* 20(4), 753–762 (2004)
9. Chang, J.H., Lee, W.S.: Finding Recent Frequent Itemsets Adaptively over Online Data Streams. In: Proc. SIGKDD (2003)
10. Cormode, G., Korn, F., Muthukrishnan, S., Srivastava, D.: Finding Hierarchical Heavy Hitters in Data Streams. In: Proc. VLDB (2003)
11. Cheng, J., Ke, Y., Ng, W.: A Survey on Algorithms for Mining Frequent Itemsets over Data Streams. *Knowledge and Information Systems* 16(1), 1–27 (2006)
12. Cormode, G., Muthukrishnan, S.: What's Hot and What's Not: Tracking Most Frequent Items Dynamically. In: Proc. PODS (2003)
13. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.: Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In: Proc. AAAI/MIT (2003)

14. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery* 17, 55–86 (2007)
15. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. In: *DMKD* (2004)
16. Jin, R., Agrawal, G.: An Algorithm for In-Core Frequent Itemset Mining on Streaming Data. In: *Proc. ICDM* (2005)
17. Jin, C., Qian, W., Sha, C., Yu, J.X., Zhou, A.: Dynamically Maintaining Frequent Items Over A Data Stream. In: *Proc. CIKM* (2003)
18. Kevin, S., Ramakrishnan, R.: Bottom-Up Computation of Sparse and Iceberg CUBEs. In: *Proc. SIGMOD* (1999)
19. Koh, J.-L., Shin, S.-N.: An Approximate Approach for Mining Recently Frequent Itemsets from Data Streams. In: Tjoa, A.M., Trujillo, J. (eds.) *DaWaK 2006*. LNCS, vol. 4081, pp. 352–362. Springer, Heidelberg (2006)
20. Leung, C.K., Khan, Q.I.: DSTree: A Tree Structure for the Mining of Frequent Sets from Data Streams. In: *Proc. ICDM* (2006)
21. Lee, D., Lee, W.: Finding Maximal Frequent Itemsets over Online Data Streams Adaptively. In: *Proc. ICDM* (2005)
22. Manku, G.S., Motwani, R.: Approximate Frequency Counts over Streaming Data. In: *Proc. VLDB* (2002)
23. Mozafari, B., Thakkar, H., Zaniolo, C.: Verifying and Mining Frequent Patterns from Large Windows over Data Streams. In: *Proc. ICDE* (2008)
24. Mao, G., Wu, X., Zhu, X., Chen, G.: Mining Maximal Frequent Itemsets from Data Streams. *Journal of Information Science* 33(3), 251–262 (2007)
25. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beerl, C., Bruneman, P. (eds.) *ICDT 1999*. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
26. Padmanabhan, B., Tuzhilin, A.: On Characterization and Discovery of Minimal Unexpected Patterns in Rule Discovery. *IEEE Transactions on Knowledge and Data Engineering* 18(2), 202–216 (2006)
27. Teng, W., Chen, M., Yu, P.S.: A Regression-Based Temporal Pattern Mining Scheme for Data Streams. In: *Proc. VLDB* (2003)
28. Tao, F., Murtagh, F., Farid, M.: Weighted Association Rule Mining using Weighted Support and Significance Framework. In: *Proc. SIGKDD* (2003)
29. Woo, H.J., Lee, W.S.: estMax: Tracing Maximal Frequent Itemsets over Online Data Streams. In: *Proc. ICDM* (2007)
30. Li, H., Lee, S., Shan, M.: Online Mining(Recently) Maximal Frequent Itemsets over Data Streams. In: *Proc. RIDE* (2005)
31. Gouda, K., Zaki, M.J.: Efficiently Mining Maximal Frequent Itemsets. In: *Proc. ICDM* (2001)
32. Bayardo, R.J.: Efficiently Mining Long Patterns from Databases. In: *Proc. SIGMOD* (1998)
33. Agarwal, R.C., Aggarwal, C.C., Prasad, V.V.V.: Depth First Generation of Long Patterns. In: *Proc. SIGKDD* (2000)
34. Burdick, D., Calimlim, M., Gehrke, J.: MAFIA: A Maximal Frequent Itemsets Algorithm for Transactional Databases. In: *Proc. ICDE* (2001)
35. Yang, G.: The Complexity of Mining Maximal Frequent Itemsets and Maximal Frequent Patterns. In: *Proc. SIGKDD* (2004)
36. Yu, J.X., Chong, Z., Lu, H., Zhou, A.: False positive or false negative: Mining frequent itemsets from high speed transactional data streams. In: *Proc. VLDB* (2004)
37. Sun, X., Orlowska, M.E., Li, X.: Finding frequent itemsets in high-speed data streams. In: *Proc. SDM 2006* (2006)

A Graph Enrichment Based Clustering over Vertically Partitioned Data

Khalid Benabdeslem, Brice Effantin, and Haytham Elghazel

University of Lyon, F-69622 Lyon, France

University of Lyon 1 - GAMA EA4608, Villeurbanne

{kbenabde,brice.effantin-dit-toussaint,elghazel}@univ-lyon1.fr

Abstract. Several researchers have illustrated that data privacy is an important and inevitable constraint when dealing with distributed knowledge discovery. The challenge is to obtain valid results while preserving this property in each related party. In this paper, we propose a new approach based on enrichment of graphs where each party does the cluster of each entity (instance), but does nothing about the attributes (features or variables) of the other parties. Furthermore, no information is given about the clustering algorithms which provide the different partitions. Finally, experiment results are provided for validating our proposal over some known data sets.

Keywords: Distributed clustering, Graph, Privacy-preserving data.

1 Introduction

Data mining is the process of extracting patterns from data and transforming it into knowledge. It deals with a large field of research with some known problems which are related to data: their huge quantity, high dimensionality and complex structure. Furthermore, data can present an additional and important problem concerning their availability and privacy. It talks about privacy-preserving data mining (PPDM) which became an active area of research in the data mining community since privacy issue gains significance and importance [10]. The problem becomes more important because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information [2]. PPDM finds numerous applications in surveillance which are naturally supposed to be “privacy-violating” applications. The key is to design methods which continue to be effective, without compromising privacy. A number of techniques have been discussed for biosurveillance, facial identification, customer transaction analysis, and identity theft. More detailed discussions on some of these issues may be found in [16,23].

The last ten years have seen extensive work on PPDM. The problem has been discussed in multiple communities such as the database community [6,7,20], the statistical disclosure control community [3,11] and the cryptography community [18]. In some cases, the different communities have explored parallel lines of work which are quite similar.

Most methods for privacy computations use some form of transformation on the data in order to perform the privacy preservation. Typically, such methods reduce the granularity of representation in order to reduce the privacy. This reduction in granularity results in some loss of effectiveness of data management or mining algorithms. This is the natural trade-off between information loss and privacy. Some examples of such techniques are as follows:

- ***The randomization model:*** The randomization method is a technique for privacy-preserving data mining in which noise is added to the data in order to mask the attribute values of records [3]. The noise added is sufficiently large so that individual record values cannot be recovered. Hence, techniques are designed to derive aggregate distributions from the perturbed records. Subsequently, data mining techniques can be developed in order to work with these aggregate distributions.
- ***The k -anonymity model and l -diversity:*** The k -anonymity model [12] was developed because of the possibility of indirect identification of records from public databases. This is because combinations of record attributes can be used to exactly identify individual records. In the k -anonymity method, we reduce the granularity of data representation with the use of techniques such as generalization and suppression. This granularity is reduced sufficiently that any given record maps onto at least k other records in the data. The l -diversity model [14] was designed to handle some weaknesses in the k -anonymity model since protecting identities to the level of k -individuals is not the same as protecting the corresponding sensitive values, especially when there is homogeneity of sensitive values within a group.
- ***Distributed privacy preservation:*** In many cases, individual entities may wish to derive aggregate results from data sets which are partitioned across these entities. Such partitioning may be horizontal (when the records are distributed across multiple entities) [5,13,27] or vertical (when the attributes are distributed across multiple entities) [25,10,28]. While the individual entities may not desire to share their entire data sets, they may consent to limited information sharing with the use of a variety of protocols. The overall effect of such methods is to maintain privacy for each individual entity, while deriving aggregate results over the entire data.

In this paper, we work in the most general setting, where we assume the data is arbitrarily partitioned between several databases. This means that there is no assumption neither on how the attributes of the data are distributed among the parties (and in particular, this subsumes the case of vertically partitioned data) nor on the clustering algorithms applied in the different parties. We propose a new approach which proceeds under privacy constraints, where the clustering algorithms are unknown, different and provide different partitions with different number of clusters.

2 Problem Statement

In this paper, we assume that there are r parties possess their private databases respectively. They want to get the common benefit for doing clustering analysis in the joint databases. For the privacy concerns, they need a private preserving system to execute the joint clustering algorithm analysis. The concern is solely that feature values associated with an individual data entity are not released.

Let r be the number of parties, each having different attributes for the same set of entities. N is the number of the common entities. Each party Ω_i ($i = 1..r$) is then described by $N \times |F_i|$ matrix where F_i is the feature subset of Ω_i . The privacy constraints that we assume are:

- $\forall i, F_i$ is only known by Ω_i and not by any other Ω_j ($i \neq j$),
- $\forall i, \Omega_i$ provides a partition P_i with an unknown clustering algorithm A_i ,
- $\forall (i, j); i \neq j; P_i$ and P_j could have different number of clusters K_i and K_j , respectively.

Initially, one party Ω_m has a partition P_m done with local known features F_m . The problem aims to update P_m by the other parties using just their clustering results P_j without any other information. So, P_m is considered as a main partition to be updated by the other secondary ones P_j under a privacy constrained framework. Subsequently, the challenge is to know (1) how to preserve privacy of data, independently of the clustering algorithm properties, (2) when an entity $x \in \Omega_m$ could or not move from its cluster in P_m because of the other partitions (P_j) and (3) how we can maintain the partition of the main party when a secondary one represents a noise. To deal with these issues, we propose a graph based framework which will be presented in the next section.

3 Graph Enrichment Based Clustering Approach

3.1 Graph Construction

In this section, we introduce our algorithm to solve the graph enrichment problem. It works to improve the partition of a main party, denoted P_m , by considering the partitions of secondary parties $\{P_1, P_2, \dots, P_r\} \setminus \{P_m\}$, under a privacy preserving data scenario. In this context, only cluster labels are shared between parties and the main one has only access to its data records for cluster analysis. Our algorithm approaches the problem by first transforming the clustering partitions (main and secondary ones) into a graph representation.

Construction 1. The data in the main party can be considered as a *complete edge-weighted graph* $G_m = \{V(G_m), E(G_m)\}$ where $V(G_m)$ ($|V(G_m)| = N$) is the vertex set and $E(G_m)$ is the edge set. Vertices in G_m correspond to data samples (entities) and edge-weights reflect similarity between pairs of linked vertices. The graph G_m is represented with the corresponding weighted similarity matrix, an $(N \times N)$ symmetric matrix $W_m = \{w_m(e) | e = (v_i, v_j) \in E(G_m)\}$ initially computed according to the features of the main party subspace.

The main partition P_m consists of K_m clusters $C_{m,1}, C_{m,2}, \dots, C_{m,K_m}$. Since a clustering is a partitioning that maximizes intra-similarities and minimizes inter-similarities among clusters, edges between two vertices within one cluster are trivially small weighted (denoting high similarity), and those between vertices from two clusters should be trivially large weighted (low similarity). Subsequently, a threshold θ could be derived from the clustering P_m , as a real number such that :

$$\theta = \min\{w_m(e) | e \in C_{m,i}, 1 \leq i \leq K_m\}. \tag{1}$$

The data in the main party can be now depicted by an *inferior threshold graph* in which vertices correspond to data samples and edges correspond to similarities which are lower than θ . In other words, the *inferior threshold graph* (termed as similarity graph in the sequel) is given by $\{V(G_m)\}$ as vertex set and $E_{<\theta}(G_m) = \{e = (v_i, v_j) | w_m(e) < \theta\}$ as edge set.

As an illustration, Table 1 gives the similarity matrix for a data set of 9 entities. Let P_m be a main partition having four clusters, $C_1 = \{x_2\}$, $C_2 = \{x_1, x_4\}$, $C_3 = \{x_3, x_5, x_7, x_8, x_9\}$ and $C_4 = \{x_6\}$. The threshold $\theta = 0.85$ is deduced according to equation 1. Figure 1 (left) shows the similarity graph obtained from Table 1 and θ .

Table 1. A weighted similarity matrix from a complete graph

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_1	0								
x_2	0.80	0							
x_3	0.90	0.70	0						
x_4	0.90	0.80	0.75	0					
x_5	0.80	0.80	0.90	0.60	0				
x_6	0.80	0.80	0.80	0.75	0.35	0			
x_7	0.85	0.90	0.85	0.90	0.90	0.25	0		
x_8	0.90	0.80	0.90	0.90	0.95	0.95	0.95	0	
x_9	0.60	0.92	0.85	0.85	0.85	0.85	0.85	0.85	0

Construction 2. Let $G_s = \{V(G_s), E(G_s)\}$ be the unweighted graph representing the partition of a secondary party Ω_s . Then, $V(G_s) = V(G_m)$. Let P_s be a secondary party with K_s clusters $C_{s,1}, C_{s,2}, \dots, C_{s,K_s}$. An edge (a, b) is added in G_s for any pair of vertices $a \in C_{s,i}$ and $b \in C_{s,j}$, with $1 \leq i \neq j \leq K_s$. As an illustration, let suppose a secondary partition P_s with three clusters, $C_1 = \{x_1, x_4, x_6\}$, $C_2 = \{x_2, x_7, x_8\}$ and $C_3 = \{x_3, x_5, x_9\}$. Figure 1 (right) shows the graph G_s obtained from P_s .

Subsequently, the graph G_m is weighted since the main party can compute the similarities from the data, while the graphs G_s are not weighted and only give representations of the partitions for the parties Ω_s .

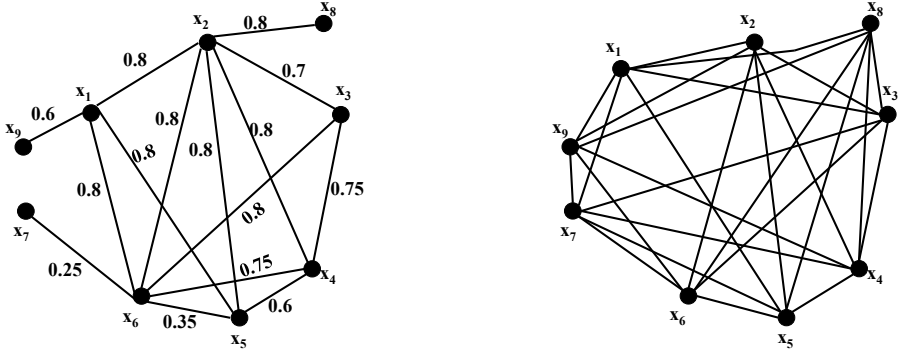


Fig. 1. Graph construction: similarity graph of partition P_m with $\theta = 0.85$ (left) and secondary graph G_s of partition P_s (right)

3.2 Graph Updating

The objective for the main party is to improve its graph in order to obtain a best partition. For that, we propose to sequentially enrich its graph by the ones issued from the secondary parties. The principle is to update the weight (w_m) of any edge, in the main party graph, by considering its presence (or not) in the secondary party graph. Thus, for any pair of vertices, we verify if the edge between them exists (or not) in the graph G_m and in the graph G_s . If an edge appears (or not) in both graphs, we consider that its presence (or not) must be confirmed and then its weight must be updated in order to increase its difference from the threshold θ . However, if an edge appears in G_m and does not in G_s (or vice versa), its weight must be updated by reducing the margin between its weight and θ .

For that, we propose to use a fitness function (eq. [2](#)), similar to the one initially proposed in [17](#) for ensemble feature selection. Given a main party and a secondary party graphs G_m and G_s , the updated weight $w_m(e)$ for each edge $e \in G_m$ is given by the following formula:

$$w_m(e) = \alpha w_m(e) + (1 - \alpha)w \quad (2)$$

where w corresponds to a weight score computed on the main party graph and used to increase (or decrease) the weight $w_m(e)$ as described before. α ($0 \leq \alpha \leq 1$) is a control parameter that is used to balance between $w_m(e)$ and w .

We define the weight score w as follows: If the edge e exists (resp. does not exist) in both graphs G_m and G_s , then w is given by the minimum (resp. maximum) of $w_m(e)$ and μ^- (resp. μ^+) given by the average weight of edges in $E_{<\theta}(G_m)$ (resp. $E(G_m) \setminus \{E_{<\theta}(G_m)\}$). μ^- (resp. μ^+) will be used to more reduce (resp. increase) the weight $w_m(e)$. The aim here is to more differentiate its difference from θ and then confirm the presence (resp. absence) of e in G_m . Otherwise, if the edge e exists (resp. not exists) in G_m and does not exist (resp.

exists) in G_s , then w is given by μ^+ (resp. μ^-) to increase (resp. reduce) the weight $w_m(e)$ allowing to e to be changed.

After $w_m(e)$ for all edges have been updated for a given value of α , the edge set $E_{<\theta}(G_m)$ may change since some edges can be added while other can be removed. These insertions or deletions mainly concern the edges with a weight closer (upper or lower) to the threshold θ . We refer to them by limit edges. Thus, the secondary graphs, considered sequentially, allow to confirm or infirm the first decision (of the main party) to keep or not these edges into the set $E_{<\theta}(G_m)$. The control parameter α is used then to balance between edge information from the main and secondary partitions. Algorithm 1 details this graph enrichment approach.

Algorithm 1. Graph updating

Input: *The main and the secondary graphs G_m and G_s*

1) From the graph G_m , we deduce:

- θ : the threshold of G_m ,
- μ^- : the average weight of edges in $E_{<\theta}(G_m)$,
- μ^+ : the average weight of edges in $E(G_m) \setminus \{E_{<\theta}(G_m)\}$.

2) *Updating*

for each edge $e \in E(G_m)$ **do**

$w_m(e) = \alpha w_m(e) + (1 - \alpha)w$, where $0.0 \leq \alpha \leq 1.0$, and w is a weight determined as follows:

if $e \in E_{<\theta}(G_m)$ and $e \in E(G_s)$ **then**

$w = \min\{w_m(e), \mu^-\}$

end if

if $e \notin E_{<\theta}(G_m)$ and $e \notin E(G_s)$ **then**

$w = \max\{w_m(e), \mu^+\}$

end if

if $e \in E_{<\theta}(G_m)$ and $e \notin E(G_s)$ **then**

$w = \mu^+$

end if

if $e \notin E_{<\theta}(G_m)$ and $e \in E(G_s)$ **then**

$w = \mu^-$

end if

end for

Output: *Updated G_m with a new weighted adjacency matrix W_m .*

It is important to note that this algorithm is applied dynamically with the receipt of a secondary partition. The main party does not wait for receiving all the possible secondary partitions, but it updates its graph when a second party sent it its partition.

Note that the updating step in Algorithm 1 is computed in time $O(N^2)$ since a new weight is computed for each edge of a complete graph on N vertices.

3.3 Graph Partitioning

The data to be clustered for the main party are depicted by a similarty graph. Thus, the clustering problem can be formulated as a graph partitioning problem.

Several feasible approaches are introduced in the graph literature to solve the graph partitioning problem for data clustering [9,26,11,21]. They basically consist in finding combinatorial structures within the similarity/dissimilarity graph. In this study, we adopt spectral clustering approach in our scheme, with especially the algorithm in [21].

In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms. This technique explores the eigenstructure of a similarity matrix to partition samples (entities) into disjoint clusters, while considering samples in the same cluster having high similarity and samples in different clusters having low similarity. The spectral clustering can be viewed as a graph partitioning task. For that, it requires the construction of a weighted graph that encodes the similarity between data samples. Vertices in the graph correspond to data samples; the weight of the edge between two vertices is a function of the similarity between the corresponding two data samples. The simplest and most straightforward way to construct a partition of this similarity graph is to solve the Min-cut problem.

In our scheme we update a similarity matrix from the procedure explained in Algorithm 1. This matrix is then presented as an input of the Algorithm 2.

Algorithm 2. Spectral clustering

Input: The new weighted adjacency matrix W_m obtained from Algorithm 1, k the number of clusters to construct, here $k = K_m$.

- 1) Compute the unnormalized Laplacian L ($L = D - W_m$, D is the diagonal matrix with $D_{ii} = \sum_{j=1}^N w_m(i, j)$)
 - 2) Compute the first k generalized eigenvectors u_1, \dots, u_k of the generalized eigenproblem $Lu = \lambda Du$.
 - 3) Let $U \in \mathbb{R}^{N \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns.
 - 4) For $i = 1..N$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i^{th} row of U
 - 5) Cluster the points $(y_i)_{i=1, \dots, N}$ in \mathbb{R} with the k -means algorithm [15] into clusters $C_{m,1}, \dots, C_{m,k}$.
-

Note that this algorithm uses the generalized eigenvectors of L , which corresponds to the eigenvectors of the matrix $D^{-1}L = I - D^{-1}W_m$ (I is the identity matrix). So in fact, the algorithm works with eigenvectors of the normalized Laplacian L , and hence is called normalized spectral clustering.

4 Experimental Results

In this section, six benchmark data sets (*Glass*, *Lung*, *Soybean*, *Wisconsin*, *Wave* from [4], and *Lsun* from [24]) were selected to test the performance of our approach. Their characteristics are given in Table 2.

The evaluation of our approach was broken down into two parts: 1) effect of the control parameter α on the quality of the returned graph enrichment based

Table 2. Used data sets

<i>Data sets</i>	<i>#instances</i>	<i>#features</i>	<i>#labels</i>	<i>Section</i>
GLASS	214	9	7	4.1
LUNG	32	56	3	4.1
LSUN	400	2	3	4.1
SOYBEAN	290	35	15	4.1
WISCONSIN	699	9	2	4.1
WAVE	5000	40	3	4.2

clustering partition, and 2) impact of adding noise partitions on enrichment performance.

In our experiments, the Rand Index [\[19\]](#) was used to measure the accuracy of a clustering solution and the *normalized spectral clustering* [\[21\]](#) was adopted as the “basis” clustering algorithm.

4.1 Quality of the Obtained Partition

First, the quality of the updated main partition using our graph enrichment based clustering approach was studied. For each data set, we created three disjoint partial views (through random selection of a third of features). The *spectral clustering* was applied to generate the partitions (main and secondary). To demonstrate the effectiveness of our graph enrichment based clustering approach, the accuracy of the obtained clustering solution was compared with those of the a) initial main partition, b) both secondary partitions, c) partition obtained on the whole data set (for which the *spectral clustering* algorithm has access to all features) and d) partition obtained using the *consensus evidence accumulation technique* proposed in [\[8\]](#). The latter consists in applying *spectral clustering* algorithm on the co-association similarity matrix, which is obtained by simply counting the fraction of clusters shared by each pair of samples in the main partition and the secondary ones.

In order to test the effect of the control parameter α on the final solution, the strategy explained before was repeated through increasing its value from 0.0 to 1.0. The results of these experiments are reported in [Figure 2](#).

The following phenomena can be observed from this figure. First of all, the spectral clustering performed on our updated similarity matrix is able to improve the clustering quality of the main partition and can achieve better solutions when compared with the consensus clustering approach. This confirms the effectiveness of our new similarity matrix to quantify the good proximity between data samples in the context of vertically partitioned data when compared to the co-association matrix in [\[8\]](#).

Another interesting phenomenon observed from [Figure 2](#) is that the clustering on all features is not always better than our privacy one. For example, our approach outperforms the clustering on all features for *Glass*, *Soybean* and *Lung* data sets, while consensus clustering approach tends to be poorer than the clustering on all features, as reported in [\[22\]](#). In our opinion, this is the most

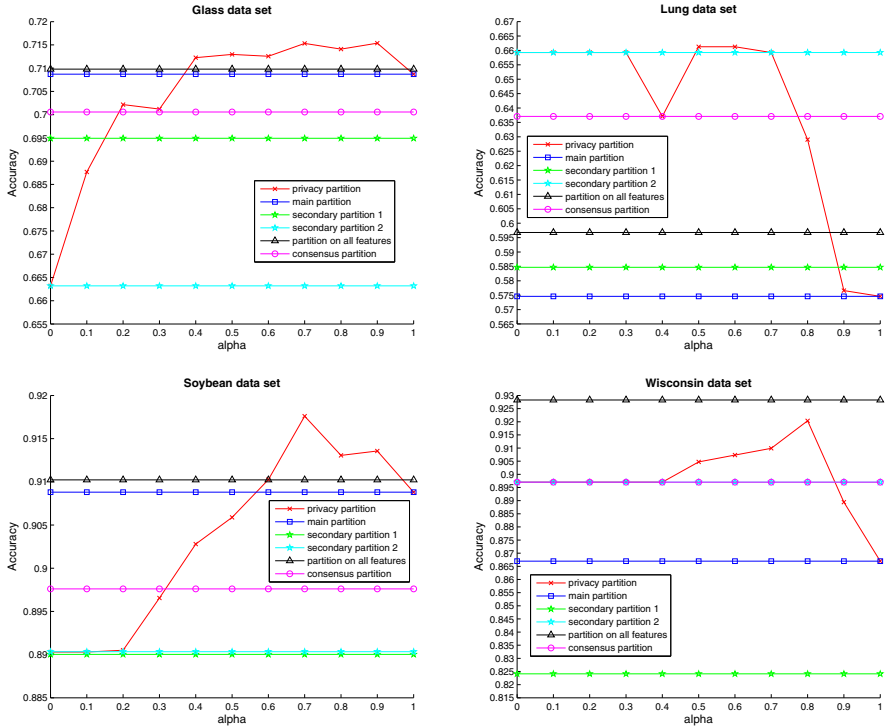


Fig. 2. Clustering accuracy under different control parameters

interesting result of this experiments since both combiner (consensus and our one) have no feature information. A good example of this observation could be given by *Lsun* data set shown in Figure 3. Remark in this figure that at first glance the two dimensional problem cannot be decomposed into two one-dimensional problems. In fact, the projection over the second dimension (Y) provides two clusters. Moreover, the projection over the first dimension (X) the structure of data is totally lost (no clusters). Hence, it is a real challenge to obtain three clusters by combining the two partitions issued from X and Y respectively. The problem seems to be exacerbated by higher dimensionality.

In our experiments on this data set, the main partition has been obtained on X 's feature (looking solely at the horizontal axis) and the secondary partition on Y 's feature (looking solely at the vertical axis). From the two dimensions point of view, it is clear that the difference in the vertical axis dominates (we note that the clustering accuracy of the secondary partition is better than the main partition) but the partitioning problem with 3 clusters appears difficult. However, these clusters have been obtained using our approach (the rand index is equal to 1 for different values of α from 0.5 to 0.8) (Figure 2 (*Lsun* data set)) which is a very interesting issue.

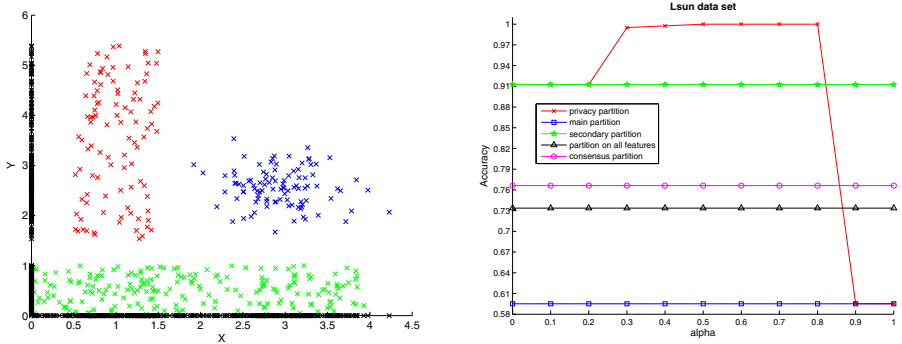


Fig. 3. *Lsun* data set

The other phenomenon observed from Figure 2 is that the performance of our method is closely related with the value of the control parameter α . It should be noted that this parameter is used to balance between edge information from the main and secondary partitions for obtaining a good clustering. The value 1.0 of the control parameter α means that only main partition is employed for similarity matrix updating, while the value 0.0 of the control parameter α represents that only the edges from the secondary partition are considered for the update. As observed from curves in Figure 2, values of the control parameter in $[0.6, 0.9]$ could achieve better accuracies. However, these results tell us that a good enrichment should draw a good balance between edge information where different data sets have different good control parameters.

4.2 Effect of Noise on the Enrichment Performance

The *Wave* data set has 40 variables where the last 19 ones are totally irrelevant with mean 0 and variance 1. We conducted several experiments on this data set in order to study the impact of adding noise partitions (as a secondary partition) on the performance of our privacy clustering approach.

The *spectral clustering* was used again as the basis clustering algorithm. We first performed the main clustering on the original 21 first variables; then a secondary partition obtained on the 19 irrelevant variables were added sequentially to update the main partition using our approach and the procedure was repeated five times. At each step, the clustering quality of our partition using three values of the control parameter α (0.7, 0.8 and 0.9 according the results in Section 4.1) was compared to the one returned by the consensus clustering approach in [8]. We report the results using Rand index in Figure 4.

The following conclusions can be drawn from these experiments: (1) the performance of the consensus clustering deteriorated markedly after the first insertion of noise partition, (2) although the quality of our clustering solution still breaks down, its robustness to noise partitions is confirmed compared to the consensus approach, especially for the first steps. However, this robustness is closely related with the value of the control parameter α .

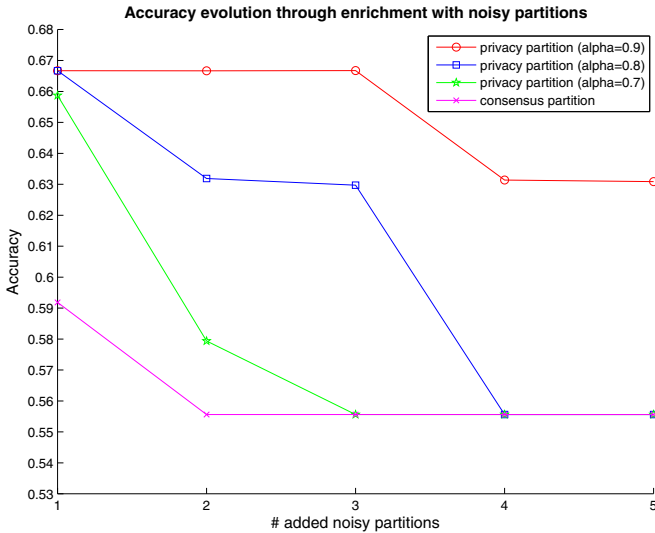


Fig. 4. Accuracy vs. the number of irrelevant secondary partitions for *Wave* data set

5 Conclusion

In this paper we proposed a new graph based approach for privacy preserving clustering. We discussed this approach for both, distributed privacy-preserving mining and handling vertically partitioned data. The key assumption in our proposal is that a privacy-preserving system should be composed of parties which do not communicate any information to each other. In other words, each party knows only its attributes and its clustering algorithm. To deal with this problem of privacy constraint, we developed a new framework on three steps. First, we modeled all partitions by graphs since the data are not communicated to each other. Second, we proposed a new algorithm for updating one graph (main partition) by another one (secondary partition) and finally, we used spectral clustering on the graph resulting from the combination. Applying this graph based strategy on some known data sets, we showed that our approach outperforms the clustering on all features. It is also better than the consensus paradigm and clearly robust to the noise.

References

1. Aggarwal, C.C., Yu, P.S.: On variable constraints in privacy preserving data mining. In: Proceedings of ACM SIAM Data Mining Conference (2005)
2. Aggarwal, C.C., Yu, P.S.: In: Privacy-Preserving Data Mining: Models and Algorithms. Springer, Cambridge (2008)
3. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of ACM SIGMOD Conference (2000)

4. Blake, C.L., Merz, C.J.: Uci repository of machine learning databases (1998)
5. Clifton, C., Kantarcioglu, M., Lin, X., Zhu, M.: Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations* 4(2) (2002)
6. Elmagarmid, A., Bertino, E., Saygin, Y., Dasseni, E.: Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering* 16(4) (2004)
7. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy-preserving mining of association rules. In: *Proceedings ACM KDD Conference* (2002)
8. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 27(6), 835–850 (2005)
9. Guenoche, A., Hansen, P., Jaumard, B.: Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification* 8, 5–30 (1991)
10. Han, S., Ng, W.-K.: Privacy-Preserving Self-Organizing Map. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) *DaWaK 2007*. LNCS, vol. 4654, pp. 428–437. Springer, Heidelberg (2007)
11. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. *Information Processing Letters* 76, 175–181 (2000)
12. Jiang, W., Clifton, C.: Privacy-preserving distributed k -anonymity. In: Jajodia, S., Wijesekera, D. (eds.) *Data and Applications Security 2005*. LNCS, vol. 3654, pp. 166–177. Springer, Heidelberg (2005)
13. Kantarcioglu, M., Vaidya, J.: Privacy-preserving naive bayes classifier for horizontally partitioned data. In: *IEEE Workshop on Privacy-Preserving Data Mining* (2003)
14. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkatasubramanian, M.: l -diversity: Privacy beyond k -anonymity. In: *Proceedings of ICDE Conference* (2006)
15. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: *Proceeding of the Fifth Symposium on Math, Statistics Ans Probability*, pp. 281–297 (1967)
16. Newton, E., Sweeney, L., Malin, B.: Preserving privacy by de-identifying facial images. *IEEE Transactions on Knowledge and Data Engineering* (2005)
17. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research* 11, 169–198 (1999)
18. Pinkas, B.: Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explorations* 4(2) (2002)
19. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66, 846–850 (1971)
20. Rizvi, S., Haritsa, J.: Maintaining data privacy in association rule mining. In: *Proceedings VLDB Conference* (2002)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
22. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
23. Sweeney, L.: Privacy technologies for homeland security. In: *Testimony before the Privacy and Integrity Advisory Committee of the Department of Homeland Security*, Boston, MA. Springer, Heidelberg (2005)
24. Ultsch, A.: Clustering with som: U^*c . In: *Proceedings of the Workshop on Self-Organizing Maps*, pp. 75–82 (2005)
25. Vaidya, J., Clifton, C.: Privacy-preserving k -means clustering over vertically partitioned data. In: *Proceedings of SIGKDD conference* (2003)

26. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(11), 1101–1113 (1993)
27. Yu, H., Jiang, X., Vaidya, J.: Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In: *Proceedings of SAC Conference* (2006)
28. Yu, H., Vaidya, J., Jiang, X.: Privacy-Preserving Svm Classification on Vertically Partitioned Data. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) *PAKDD 2006*. LNCS (LNAI), vol. 3918, pp. 647–656. Springer, Heidelberg (2006)

A Method for Finding Groups of Related Herbs in Traditional Chinese Medicine

Lidong Wang^{1,2}, Yin Zhang¹, Baogang Wei¹, Jie Yuan¹, and Xia Ye²

¹ College of Computer Science and Technology, Zhejiang University, Hangzhou, China

² Hangzhou Normal University, Hangzhou, China, 310012

{Violet_wld, java_mc}@163.com, {yinzhang, wbg}@zju.edu.cn,
yexia810128@yahoo.com.cn

Abstract. As a complementary system to Western medicine, Traditional Chinese Medicine (TCM) provides a unique theoretical and practical approach of treatment to diseases over thousands of years. Accompanying with the increasing number of TCM digital books in digital library, there is an urgent need to explore these resources by the techniques of knowledge discovery. We present a method for creating a network of herbs and partitioning it into groups of related herbs. The method extracts structured information from several TCM digital books, then a new method named Support and Dependency Evaluation (SDE) is presented for herbal combinational rule mining. The herbal network is created from the extracted dataset of paired herbs. The partitioning procedure is designed to extend FEC algorithm to deal with the weighted herbal network. Experiments demonstrate that the method proposed has the capability of discovering groups of related herbs.

Keywords: Traditional Chinese Medicine (TCM), SDE, herbal combinational rule, group detection, FEC.

1 Introduction

Traditional Chinese Medicine (TCM) has carried a big weight in the health care for Chinese people, accompanying with a public interest in foreign countries[1]. Increasing users intend to learn TCM domain knowledge from digital books, so the automatic analysis of TCM digital books becomes useful. There are about 10,000 currently known herbs and a large number of related digital books in digital library (DL). A comprehensive study of a formula involving several herbs might require a researcher to be familiar with many digital articles. Merely locating all relevant books in a database by using a simple search would be inefficient and time consuming.

Compared with western biomedicine, countless TCM practices in thousands of years accumulate numerous cases of combinatorial medicines as the form of formulae. TCM contains four data types, which are herb, formula, symptom and syndrome. The knowledge discovery research in TCM mainly adapts to exploring the relationship between those of four data types, such as syndrome differentiation[10], herbal combination rule[4], modeling of inquiry diagnosis[4], formula-syndrome relationship

mining[5] , syndrome-gene relationship discovery[9]so on. Shi [2] presents a novel search engine called Msuggest to provide the diverse semantic recommended herbs and book pages when a reader searching for herbal information in digital library. However, the recommended herbs from Msuggest are extracted through word co-occurrence in medicine dictionary, without through employing knowledge discovery techniques. Actually, the users want to learn quickly and effectively through the contents of each book, so we hope to extract the hidden knowledge from TCM digital book in our DL[2].

In this article, we present a method to find groups of related herbs based on herbal combinational rule mining. Combination of herbs can increase their medical effectiveness, or reduce the side effects of certain herb. The method creates a network of herbs from TCM digital books and partitions this network into groups. The herbs in each group are functionally related. This method can thus be a valuable tool that both summarizes available information and indicates some unknown interactional herbs. The groups thereby imply connections among herbs may be overlooked or that would require much time and effort to be found manually¹.

2 Method Overview

For researchers in TCM direction, it is impractical to search different herbs and their combinational rules manually through literatures. The online digital books present a gold mine of available information. In fact, the ability extract structured information from different digital books is essential for the mining of herbal combination rule and other related mining tasks.

In the beginning, we conduct structured information extraction from OCR documents of TCM books using language processing techniques. Then a new method named Support and Dependency Evaluation(SDE) is proposed to conduct combinational rule mining by extracting related paired herbs. Network is then created from these sets of herbs. In the network, each node represents a herb, and an edge connects two herbs if they represent as related paired herbs. To find groups of related herbs, we apply the algorithm of community detection to this task.

It needs to note that group/community² detection for herbs is different from clustering, since the latter always focuses on clustering herb in terms of the same attribute, such as source, nature or efficacy. However, the groups/communities mean to imply connections among herbs, also including some herbs that may otherwise be overlooked or that would require much time and effort to be found manually. The relationship between them means that combination of two or more herbs can increase their original medical effectiveness or can eliminate the toxicity of one herb.

Despite a long history of investigation, surprisingly, there is not a single universally accepted definition of the community. Computer scientists prefer to define community as “a group of vertices in which there are more edges between vertices within the group than to vertices outside of it”[11]. In the literature, algorithms developed to detect communities can generally be divided into three main categories:

¹ It is important to note that the herbal groups in the results are not meant to perfectly reproduce reality in TCM. It simply provides a powerful method for organizing and presenting information from the literature.

² “Group” and “community” represent the same meaning in this article.

Modularity-based methods[16,17], Spectral algorithms[18], Methods based on statistical inference[19], and other alternative methods. Modularity-based methods always transform problem of the social network clustering to the optimization of predefined objective function such as modularity. However, Fortunato[12] has showed that modularity optimization may fail to identify modules smaller than a scale which depends on the total size of the network and on the degree of interconnectedness of the modules, so it is not a robust method for group detection of herbal network that may not contain large amount of nodes. Meanwhile, spectral algorithm has the limitation that it requires the prior knowledge of the number of groups, which is impossible to be obtained during our task.

Yang [13] presented a method named FEC that is demonstrated to be fast, effective for both signed network and positive network. Thus, we adopt FEC algorithm for herbal group detection with slight modification. Because the relationship between herbs varies in intensity, the weight during them should not be directly assigned as 1. Thus, the procedure for creating herbal graph should calculate the weight between each two paired herbs. To effectively find the community, we extend the FEC algorithm to deal with weighted networks.

In General, our proposed method for finding groups of related herbs contains three main phases: 1) Obtaining paired herbs dataset through combinational rule mining; 2) Creation of herbal network; 3) Group detection for herbal network. Our paper is organized as follows, in Section 3, the process of obtaining paired herbs set from digital books is introduced. The creation of herbal graph and the key technique for group detection is shown in Section 4. In Section 5, we provide discussions of experimental results, as well as utility and effectiveness of our method. Finally, we conclude with some potential extensions of our methodology in Section 6.

3 Obtaining Dataset of Paired Herbs

3.1 Data Preprocessing for TCM Digital Books

How to extract structured information that can be used directly to knowledge discovery? Our paper proposes two steps for structured information extraction. Firstly, regular expression matching is employed to extract formula-herb structured information(Table 1) from TCM digital books. Subsequently, herbal name extraction is conducted by LJparser tool[3] with the method of importing symptom name dictionary before word segmentation(see red rectangles in Table 1).

Table 1. Example of extracted formula-herb structured information

Formula	桂枝汤
Herbs	桂枝三两 (9克) 芍药三两 (9克) 炙甘草二两 (6克) 生姜三两 (9克) 大枣十二枚 (7枚)

3.2 Combinational Rules Mining

To use combinational medicines properly, one key issue is to realize the combinational rules of multiple herbs. The reason is that the potency of a single herb

is usually limited. But when two herbs are used together, they would interact with each other and display their superiority over a single herb in the treatment of diseases.

When two herbs are frequently used in combination with each other, they are more likely to be paired drugs. Therefore, the Support attribute[7] and dependency relationship could be used to discover paired herbs. We propose a method of relationship mining named SDE(Support and Dependency Evaluation) as outlined in Algorithm 1, which is defined as follows:

$$\text{support} = P(h_1, h_2) \quad (1)$$

$$D(h_1, h_2) = p(h_1, h_2) \log_2 \frac{p(h_1, h_2)}{p(h_1)p(h_2)} \quad (2)$$

$$SDE_value = \alpha \cdot \text{support} + \beta \cdot D(h_1, h_2) \quad (3)$$

We can conclude from the equations that SDE simultaneously reflects the Support attribute ($p(h_1, h_2)$) and Correlation attribute(the ratio of $p(h_1, h_2)$ to $p(h_1)p(h_2)$). The proportion of Support attribute is highlighted, although it has been included within Correlation attribute. The algorithm outputs the dependency modes under the condition of $SDE_value > threshold$. We test our algorithm by varying the threshold from 0 to 0.01 with a step size of 0.005, then select the optimum threshold with best precision performance. Meanwhile, parameters α, β are chosen as 0.5 from repeated experiments. Besides, an important step is carried out, that is, to remove Raidx Glycyrrhizae(甘草) from each dependency mode. That is because Raidx Glycyrrhizae is frequently used in all kinds of formulas to decrease or moderate medicinal side-effects and to regulate actions of all other herbs.

Algorithm 1. SDE(Support and Dependency Evaluation)

Input: $\{h_i \mid i \in [1..m]\}$ // h_i represents the i th herb

Output: paired herbs set

Method:

1. scan the components for each formula;
 2. for each (h_i, h_j)
 3. compute the joint probability of herb h_i and h_j , $p(h_i, h_j)$;
 4. end for
 5. for each (h_i, h_j)
 6. compute SDE value according to Eq.(2)~(3);
 7. end for
 8. for shreshold = 0 to 0.01
 9. Compute the precision compared with Database of Paired Drugs(DPD);
 10. end for
 11. select the shreshold according to the value of precision;
 12. if $SDE_value > threshold$ and $h_i, h_j \neq \text{Raidx Glycyrrhizae}$
 13. output the paired herbs;
 14. end if
-

4 Group Detection

4.1 Herbal Network

The creation of herbal graph from the dataset of paired herbs is performed following a well-known procedure[15]. Using a graph $G(N, E)$, where N is the number of vertices of the graph, and E is the number of edges. Each vertex in the graph represents a herb, and an edge exists between two vertices if the herbs represents paired herbs. Some paired herbs would strengthen original efficacy of single herb, such as “Rhizoma Coptidis” (黄连) and “Scutellaria Baicalensis” (黄芩); Some of them only alleviate the side effect of one herb, for example, the combination of “Pinellia ternate” (半夏) and “Gingembre” (生姜) can alleviate the toxicity of “Pinellia ternate”. Therefore, in consideration of the different strength of efficacy between herbs, we use weighted edges. The weight between herbs can also be considered the degree of influence a node has on the other node. We use this index to divide the network so that nodes which have higher influence on each other are grouped together. Rumi[14] define influence as the capacity to have an effect on some nodes. The total influence of herb i on herb j is thus dependent on the sum of all the weighted paths from herb i to herb j . Here we just consider the situation of direct link from herb i to herb j , denoted as $(i \rightarrow j)_0$, which is given by the element of adjacency matrix \mathbf{W} whose elements W_{ij} are defined as:

$$W_{ij} = \begin{cases} (SDE_value) * 100 & \text{if } \exists \text{ an edge from vertex } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Since the range of SDE_value always fall in the region of $[0,0.1]$, we normalize it to $[0, 10]$.

The resulting graph has the property of complex network[21], which is scale free. A portion of the herbal network from classical TCM digital books (see Section 5.1) is illustrated Fig.2. The property of scale free means the resulting graph has a power law distribution in its Degree, that is, the number of vertices of Degree k is given by $C_k = C * k^{-r}$, where C and r are constant, r is the power law exponent. The property of scale free is shown in Fig.1, where we plot the data on a log-log scale for herbal network. The number of vertices(y axis) is plotted against the degree of vertex(x axis) on a log-log scale. The Degree of vertices in herbal network falls in the region of $[1,17]$. The deviation from the power law for low vertex degree is typical.

Such law graph shows that the herbal network created by our method satisfies the property of complex network. Hence, we believe that the algorithm of community detection can be applied to our task.

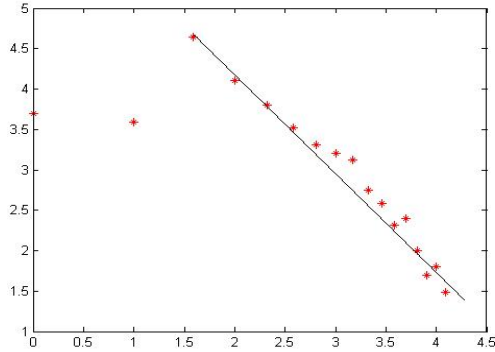


Fig. 1. The number of vertices is plotted against the degree of the vertex on a log-log scale

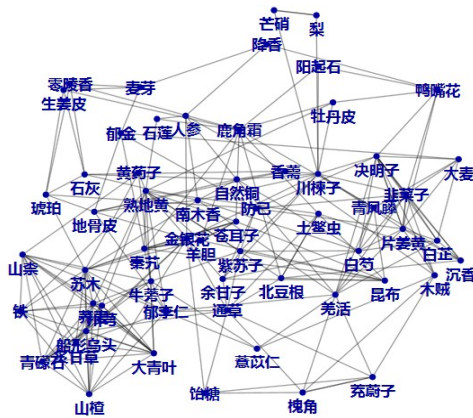


Fig. 2. Extracted herbal network with a part number of paired herbs from 《Fang Ji Ci Dian》

4.2 Partitioning the Network into Groups

There is no formal definition for a group of vertices within a network. A network can be said to have community structure if it consists of subsets of herbs, with many edges connecting herbal nodes of the same subset but few edges lying between subsets. Finding communities within a network is an efficient way to identify groups of related vertices.

As mentioned in Section 2, the community discovery process we use is based on FEC algorithm, which has been shown to identify communities with high accuracy and low running time. FEC is presented with two phases that are Finding Community(FC) and Extraction Community(EC). The latter applies a cutoff criterion to the transformed adjacency matrix and divides it into two block matrices. The former transforms the adjacency matrix to compute their transition probability vector and sorts them for each row. Our method focuses on extending the FC phase to deal with weighed networks. In the following we describe the modified FC phase.

The FC phase adopts an agent-based approach to model the problem of finding the community that contains a specific node for signed network. The agent's walk can be viewed as a stochastic process defined based on the links' attributes. When the agent arrives at a node, it will select one of its neighbors at random and then go there. Let p_{ij} be the probability of the agent walking from node i to its neighbor node j . In a weighted positive network, this probability can be computed as follow:

$$p_{ij} = \frac{W_{ij}}{\sum_{\langle i,j \rangle} W_{ij}} \quad (5)$$

where W_{ij} represents the weight of link $\langle i,j \rangle$, computed by Eq.(4). Then, let $P_t^l(i)$ be the probability that agent starting from node i can eventually arrive at a specific sink node t with l steps. The value of $P_t^l(i)$ can be estimated iteratively by

$$P_t^l(i) = \begin{cases} \sum_{\langle i,j \rangle} p_{ij} \cdot P_t^l(j) & \text{else} \\ 1 & i = t \end{cases} \quad (6)$$

Agents that start from nodes within the community of the sink node should reach the sink node more easily within l steps since more paths can be chosen. The parameter l is set to 10 in our experiment that has been found to be sufficiently well for all experiments[13]. In the contrary, agents that start from nodes outside the community of the sink node would reach the destination more difficultly. Mathematically speaking, $P_t^l(i)$ should follow the property as

$$\forall_{i \in G_t} \forall_{k \notin G_t} (P_t^l(i) > P_t^l(k)) \quad (7)$$

where G_t denotes the community where node t is situated. Finally, the procedure for finding the community can be described as follows, which is similar with the steps described in [13]:

- 1) Calculate $P_t^l(i)$ for each node i ;
- 2) Rank all the nodes according to their associated value of $P_t^l(i)$.

5 Experiments

5.1 Experiments on Combinational Rule Mining

We test on 4 classical TCM books 《Shang Han Lun》, 《Fang Ji Xue》, 《Jin Kui Yao Lue》, 《Fang Ji Ci Dian》 (《伤寒论》, 《方剂学》, 《金匱要略》, 《方剂词典》) from our digital library[6].

Firstly, we conduct experiments on threshold selecting. The experimental results of threshold selection are showed in Fig.3, with the threshold varying from 0 to 0.1 with a step size of 0.005. The precision in axis of ordinates is presented as the percentage of generated paired herbs matching with the record in Database of Paired

Drugs(DPD)³. There exists great variety in the amount of paired herbs in each book, so the threshold should be set differently according to the experimental results. With the consideration of both returned number of paired herbs and performance of precision, we select 0.045, 0.03, 0.07, 0.055 as threshold for 《Jin Kui Yao Lue》, 《Shang Han Lun》, 《Fang Ji Xue》, 《Fang Ji Ci Dian》 respectively.

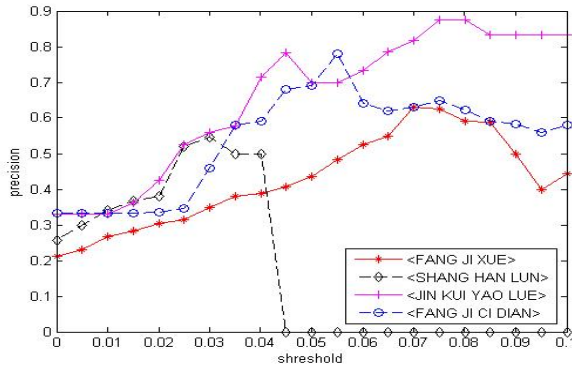


Fig. 3. Experimental results of threshold selection

The results of discovered top 10 paired herbs from book 《Shang Han Lun》 are presented in Table 2. The second column indicates whether these discovered herbs exist in DPD(Database of Paired Drugs)³. Those records with field “Yes” substantiate the existence of paired herbs. The extracted paired herbs that are included in DPD can be extracted by knowledge discovery techniques. Furthermore, many paired herbs that do not exist in DPD are revealed with high *SDE_value*, such as “Agaric Polyporus” and “Rhizoma Alismatis”, “Peach Kernel” and “Tabanus Gadfly”. With TCM experts instructions, this case is not always false because some of combinations are highly likely to be paired herbs that are worthy of further analysis and verification. For instance, “Agaric Polyporus” and “Rhizoma Alismatis” can be combined for the treatment of

Table 2. Discovered top 10 paired herbs from 《Shang Han Lun》

Discovered paired herbs	Exist in DPD	<i>SDE_value</i>
Agaric Polyporus, Rhizoma Alismatis(猪苓、泽泻)	No	0.0521
Anemarrhena Asphodeloides, Roundshaped Rice(知母、粳米)	No	0.0442
Sanguisuga, Tabanus Gadfly(水蛭、虻虫)	Yes	0.0438
Anemarrhena Asphodeloides, Gypsum Fibrosum(知母、石膏)	Yes	0.0428
Peach Kernel, Tabanus Gadfly(桃仁、虻虫)	No	0.0387
Tabanus Gadfly, Sanguisuga(桃仁、水蛭)	No	0.0387
Mangnolia officinalis, fructus aurantii immaturus(厚朴、枳实)	Yes	0.0377
Roundshaped Rice, Gypsum Fibrosum(粳米、石膏)	Yes	0.0361
Bupleurum Chinense, Scutellaria Baicalensis(柴胡、黄芩)	Yes	0.0343
Fossil Fragments, Ostracean(龙骨、牡蛎)	Yes	0.0322

³ http://www.cintcm.com/cintcm_content/tcm_database/zhongyao_yaodui.htm

“dampness-heat” syndrome, with the symptoms “difficulty in micturition and dropsy”. Meanwhile, the combination of “Peach Kernel” and “Tabanus Gadfly” has synergistic action and the efficacy of removing blood stasis and resolving accumulation. It is observed that utilizing powerful computers and efficient algorithms in TCM could also promote the development of compatibility rule research.

5.2 Experiments on Community Detection

5.2.1 Evaluation Criterion

We test our modified community mining algorithm based on FEC with the herbal network extracted from the above four classical TCM books so as to evaluate its effectiveness. To evaluate the group detection quality, we use the unsupervised metrics normalized cut(NCut). NCut is objective of the normalized cut algorithm. Given a community partition $C = \{C_1, C_2, \dots, C_k\}$, the normalized cut is defined as

$$NCut(C_1, \dots, C_k) = \sum_{i=1}^K \frac{Cut(C_i, \bar{C}_i)}{\sum_{p \in C_i} \sum_q W_{pq}} \quad (8)$$

where \bar{C}_i denotes the set of nodes that are not in C_i and $Cut(C_i, \bar{C}_i) = \sum_{p \in C_i, q \in \bar{C}_i} W_{pq}$. Obviously, the smaller the value of NCut is, the better the partitioning quality becomes.

5.2.2 Sensitivity Analysis of Sink Node Selection

During FC phase, the sink node is randomly selected. However, the sink node would influence the performance of the detection results. To illustrate the influence of different sink nodes, a benchmark social network of a US college football association in the 2000 season is used for evaluation. The network includes 115 nodes and 613 edges representing football teams and games among those teams, respectively. All the 115 teams are divided into 12 conferences. Generally, each conference naturally forms a community of the network. We select three vertices as sink node according to the rank of each vertex's Degree, which has the lowest Degree, medium Degree and highest Degree, respectively. Fig.4a shows the adjacency matrix of the football association network. Fig.4(b-d) show the FEC output of the football association network selecting three different sink nodes. As shown in Fig.4, the communities extracted by FEC varies with different sink. In Fig.4(b), 13 blocks are identified, and 11 blocks are identified in Fig.4(c). The number of the blocks mismatches the number of conference, except for the results in Fig.4(d), which is extracted by selecting the node with highest Degree as sink. It shows that selecting node with highest Degree can obtain better performance than other two kinds of sink nodes.

5.2.3 Evaluation on Herbal Network

The herbal network extracted from 4 classical TCM books is applied to the evaluation of finding groups of related herbs. The network includes 255 nodes and 1115 edges. To evaluate the effectiveness of our method, we compare the modified FEC algorithm

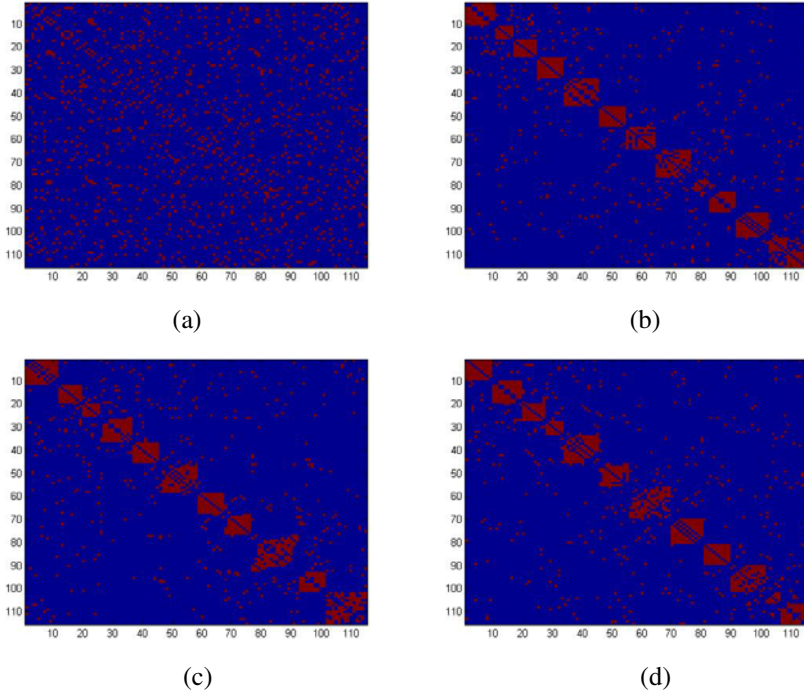


Fig. 4. Extracting communities from football social networks using FEC. (a) The adjacency matrix of football network. (b) The FEC output of football network selecting the node with lowest Degree as sink. (c) The FEC output of football network selecting the node with medium Degree as sink. (d) The FEC output of football network selecting the node with highest Degree as sink.

with other related methods, such as GN algorithm, and original FEC algorithm[13]. It is important to note that the herbal network is represented as unweighted for those of related methods. In an unweighted network, W_{ij} is set to 1 if \exists an edge from vertex i to vertex j .

As shown in Table 3, the performance shows that the extension of FEC to deal with weighted network can improve the performance of community detection for herbal network, although the improvement is not obvious. A total of 12 groups are detected in Fig.5, and each detected group is located in a block with the elements given the same color. To present the usefulness of our results, we discuss features of these communities. Used in conjunction with TCM digital books, these communities allow us to suggest connections between herbs of one group. Here we simply present two groups to demonstrate the features, as illustrated in Fig.6.

Table 3. Experimental results of different algorithms on herbal network

Methods	NCut
GN	7.87
FEC	8.43
Extension of FEC	7.84

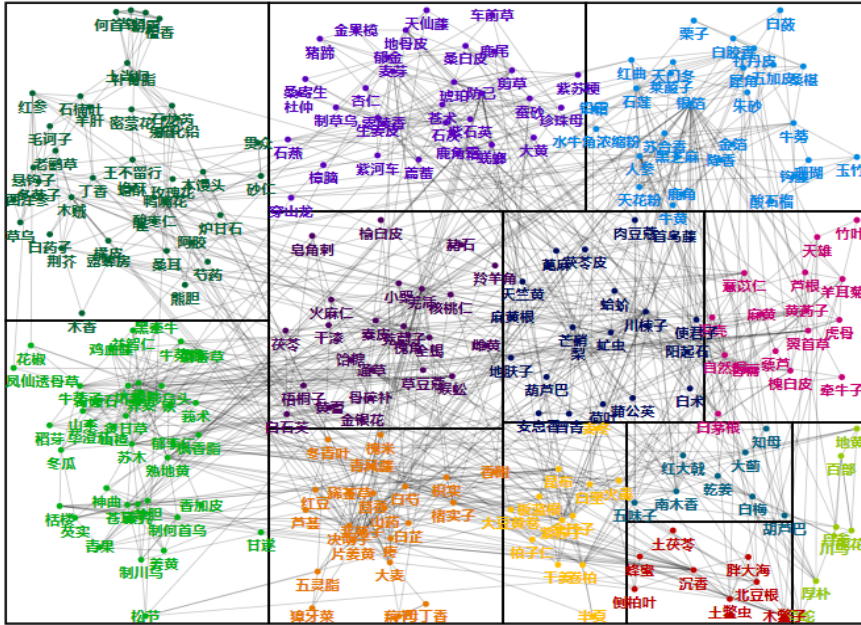


Fig. 5. Groups detected by our method from herbal network

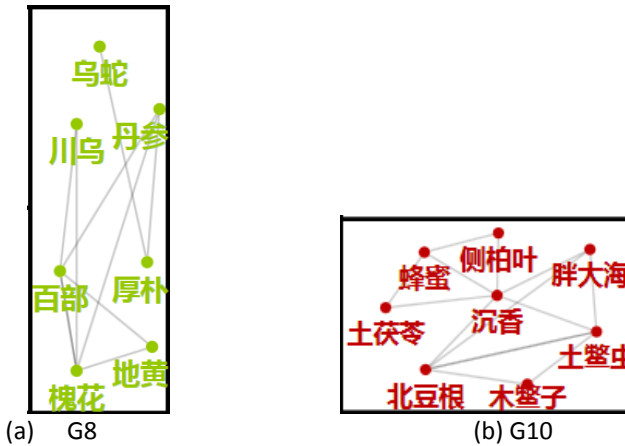


Fig. 6. Two groups detected from herbal network

The detected group G8(“Garter Snake”(乌蛇), “Mangnolia Officinalis”(厚朴), “Salvia Miltiorrhiza”(丹参), “Radix Rehmanniae”(地黄), “Radix Stemonae”(百部), “Monkshood”(川乌), “Flos Sophorae”(槐花)), contains the direct related paired herbs, such as “Mangnolia Officinalis” and “Garter Snake”, “Radix Stemonae” and “Salvia Miltiorrhiza”, “Radix Stemonae” and “Flos Sophorae” et al. However, there also exists relationship between some unconnected herbs, such as “Mangnolia Officinalis” and “Radix Rehmanniae”, “Mangnolia Officinalis” and “Radix Stemonae”. “Mangnolia Officinalis” and “Radix Rehmanniae” both have the efficacy of clearing away heat and toxic material. The combination of “Mangnolia Officinalis” and “cassia twig”(桂枝) has the efficacy of relieving cough, in the condition of severe cough, “Radix Stemonae” can be added for remedy. Additionally, we discovered that both directly connected to “Mangnolia Officinalis”, “Garter Snake” and “Salvia Miltiorrhiza” can be combined in a formula “Ge Gen Quan Xie Jiu” for activating meridians to stop pain. It would be time consuming for a researcher to ascertain this connection manually from digital books.

Here we present similar results from one other community(G10). During these four herbs, “semen momordicae”(木鳖子), “eaglewood”(沉香), “Rhizoma Menispermii”(北豆根) and “ground beetle”(土鳖虫), both of two herbs have pairwise correlation except “semen momordicae” and “eaglewood”. Actually, those two herbs are associated for the remedy of dysentery of children[23], though they are seldom combined. Similar results can also be derived from “Cacumen biotae”(侧柏叶) and “Smilax officinalis”(土茯苓). Although no literature has recorded their relation, both of these two herbs have the efficacy of restraining Sebum secretion, and have been used in cosmetics products, which demonstrate the latent relation between them.

Hence, it is valuable to conduct further research for some unconnected herbs, although our results are not meant to perfectly model reality of TCM. Similar results can also be obtained from other detected groups. In general, group detection of related herbs not only cluster some herbs with intensive connection, but also imply connections among herbs may be overlooked or that would require much time to find manually. Meanwhile, this also provides the probability of combination of other unrelated herbs to the researchers for further study, which is known as new formula discovery[22].

6 Conclusions

This paper has briefly reported how we implement knowledge discovery from TCM digital books. Structured information(such as a formula with corresponding herbs) should be extracted before data mining. Combinational rule of paired herbs and group mining of herbal networks are what we conduct in our experiments. Our paper effectively produces a list of groups of functionally related herbs by the method of SDE. The identification of communities of herbal network allow researchers find some unknown paired herbs, and all the herbs in one group with indirect relationship

would be worthy of further study. The herbs within a community are weighted, which can be considered as the degree of influence a node has on the other node in our paper. The community detection method based on FEC produces better result than other general methods.

The factor that most limits our result is the placement of man herbs in large communities in our results. Large communities are more difficult for us to discover potential paired herbs. For future work, we believe that subdivide large communities can make the results more effective. Additionally, it would be important to develop a universal structured information extraction platform for all kinds of TCM digital books. Meanwhile, the dosage information of each herb should be taken into consideration by the observing how the dosage change of some herbs could influence the curability of related syndrome and symptoms.

Acknowledgments. This work is supported by National Science Foundation of China (No.60673088), CADAL(China Academic Digital Associative Library) project , the Special Fund for Basic Scientific Research of Central Colleges, Zhejiang University. The authors are grateful to the anonymous reviewers for their careful and insightful reviews.

References

1. Honda, K., Jacobson, J.S.: Use of complementary and alternative medicine among United States adults: the influence of personality, coping strategies, and social support. *Preventive Medicine* 40, 46–53 (2005)
2. Shi, S.M., Wei, B.G., Yang, Y.: Msuggest: a semantic recommender framework for traditional chinese medicine book search engine. In: *CIKM 2009*, pp. 533–540. ACM Press, Hong Kong (2009)
3. Ling_Join Software, <http://www.lingjoin.com/download/LJParser.rar>
4. Qiao, S.J., Tang, C.J., et al.: Mining the compatibility law of multidimensional medicines based on dependence model sets. *Journal of Sichuan University (Engineering and Science Edition)* 39(4), 134–138 (2007) (in Chinese)
5. Wang, Y.Q., Yu, Z.H., Jiang, Y.G., et al.: Automatic symptom name normalization in clinical records of traditional chinese medicine. *BMC Bioinformatics* 11, 40–50 (2010)
6. China Academic Digital Associative Library., <http://www.cadal.zju.edu.cn>
7. Feng, Y., Wu, Z.H., Zhou, X.Z., et al.: Knowledge discovery in traditional chinese medicine: state of the art and perspectives. *Artificial Intelligence in Medicine* 38, 219–236 (2006)
8. He, Q.F., Cui, M., Wu, Z.H., Zhou, X.Z., Zhou, Z.: Compatibility knowledge discovery in Chinese medical formulae. *Chin. J. Inf. Tradit Chin. Med.* 11, 655–658 (2004)
9. Zhou, X.Z., Liu, B.Y., Wu, Z.H., Feng, Y.: Integrative mining of traditional Chinese medicine literature and MEDLINE for functional gene networks. *Artificial Intelligence in Medicine* 41, 87–104 (2007)
10. Liu, X., Hong, W., Song, J., Zhang, T.: Using Formal Concept Analysis to Visualize Relationships of Syndromes in Traditional Chinese Medicine. In: Zhang, D. (ed.) *ICMB 2010*. LNCS, vol. 6165, pp. 315–324. Springer, Heidelberg (2010)
11. Clauset, A.: Finding local community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* 72(2) (2005)

12. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. of the National Academy of Science* 104(1), 36–41 (2007)
13. Yang, B., Cheung, W.K., Liu, J.: Community mining from signed social networks. *IEEE Trans. on Knowledge and Data Engineering* 19(10), 1333–1348 (2007)
14. Rumi, G., Kristina, L.: Community Detection using a measure of influence. In: *Proc. SNA-KDD* (2008)
15. Stephens, M., Palakal, M., Mukhopadhyay, S.: Detecting gene relations from Medline abstracts. *Pac. Symp. Biocomput.* 6, 483–496 (2001)
16. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69(2), 26113 (2004)
17. Mei, J., He, S., Shi, G., Wang, Z., Li, W.: Revealing network communities through modularity maximization by a contraction-dilation method. *New Journal of Physics* 11(4) (2009)
18. Mitrovic, M., Tadi, B.: Spectral and dynamical properties in classes of sparse networks with mesoscopic in homogeneities. *Physical Review* 80(2), 026123 (2009)
19. Ren, W., Yan, G., Liao, X., Xiao, L.: Simple probabilistic algorithm for detecting community structure. *Physical Review* 79(3), 36111 (2009)
20. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Physical Review* 70(6) (2004)
21. Fortunato, S.: Community detection in graphs. *Physics Reports* 486, 75–174 (2010)
22. Yang, H.J., Chen, J.X., Tang, S.H., Li, Z.K., Zhen, Y.S., et al.: New drug R&D of traditional chinese medicine: role of data mining approach. *Journal of Biological Systems* 17(3), 329–347 (2009)
23. <http://baike.baidu.com/view/51584.htm>

A New Hybrid Clustering Method for Reducing Very Large Spatio-temporal Dataset

Michael Whelan, Nhien-An Le-Khac, and M.-Tahar Kechadi

School of Computer Science and Informatics, University College Dublin, Belfield,
Dublin 4, Ireland

{michael.whelan, an.lekhac, tahar.kechadi}@ucd.ie

Abstract. Spatio-temporal datasets are often very large and difficult to analyse. Recently a lot of interest has arisen towards data-mining techniques to reduce very large spatio-temporal datasets into relevant subsets as well as to help visualisation tools to effectively display the results. Cluster-based mining methods have proven to be successful at reducing the large size of raw data by extracting useful knowledge as representatives. As a consequence, instead of dealing with a large size of raw data, we can use these representatives to visualise or to analyse the data without losing important information. In this paper, we present a new hybrid approach for reducing large spatio-temporal datasets. This approach is based on the combination of density-based and graph-based clustering. Drawing on the Shared Nearest Neighbour concept, it applies the Euclidean metric distance to determine the nearest neighbour similarity. We also present and discuss the evaluation of the results for this approach.

Keywords: spatio-temporal datasets, data reduction, density-based clustering, shared nearest neighbours.

1 Introduction

Today, many natural phenomena present intrinsic spatial and temporal characteristics. Besides the applications concerned with climate change, the threat of pandemic diseases, and the monitoring of terrorist movements are some of the newest reasons why the analysis of spatio-temporal data has attracted increasing interest. Spatio-temporal datasets are often very large and difficult to analyse [1][2][3]. Although visualisation techniques are widely recognised to be powerful in analysing these datasets [4], the techniques currently used in the existing applications are not adequate for decision-support systems when used alone [5]. Data Mining (DM) techniques have been proven to be of significant value for analysing spatio-temporal datasets [6][7]. It is a user-centric, interactive process, where DM experts and domain experts work closely together to gain insight on a given problem. However, several open issues have been identified ranging from the definition of techniques capable of dealing with very large datasets to the development of effective methods for interpreting and presenting the final results. An approach for dealing with the

intractable problem of learning from huge databases is to reduce this huge set to a small subset of data for analysis [2]. It would be convenient if large datasets could be replaced by a small subset of representative patterns so that the accuracy of estimates (e.g., probability density, dependencies, class boundaries) obtained from such a reduced set should be comparable to that obtained using the entire dataset.

As there are many reduction techniques presented in literature such as sampling [8], data compression [9], scaling [10], etc., most of them are concerned with reducing the dataset size without paying attention to their spatial properties. Cluster-based methods [5][11] have been proposed as feasible approaches to reducing very large dataset by representing large groups of data by different cluster properties such as cluster centres, cluster representatives, etc. Clustering is one of the fundamental techniques in DM. It groups data objects based on the characteristics of the objects and their relationships. It aims at maximising the similarity within a group of objects and the dissimilarity between the groups in order to identify interesting structures in the underlying data. Some of the benefits of using clustering techniques to analyse spatio-temporal datasets include: (1) the visualisation of clusters can help understanding the structure of spatio-temporal datasets; (2) the use of simple similarity measures to overcome the complexity of the datasets including the number of attributes, and (3) the use of cluster representatives to help filter (reduce) datasets without losing important and interesting information. Furthermore, we want to exploit the important aspect of spatio-temporal data (i.e., objects that are physically and temporally close tend to be "similar").

In [5][11] the authors presented a cluster-based data reduction strategy that is to be incorporated in a system of exploratory spatio-temporal data mining [10][12], to improve its performance on analysing very large spatio-temporal datasets. In [11] they implemented the popular centre-based clustering method known as K-medoids. This algorithm was chosen for its simplicity, however its representatives (medoid points) cannot reflect adequately all important features of the datasets because this technique is not sensitive to the shape of the datasets (convex). The approach presented in [5] constitutes a new solution based on a combination of density-based (DBSCAN [13]) and graph-based clustering (Shared Nearest Neighbour [14]). The use of DBSCAN core (or specific core) points as representatives performed much better than centre-based representatives with regards to the shape of the data in spatial datasets. DBSCAN was combined with the Shared Nearest Neighbour (SNN) Similarity algorithm in order to deal with the problem of differences in density. In this approach, SNN degree is used to determine core points. As mentioned in [5], it is efficient for cases with small or medium variations in density of different investigating areas. However, it could be an issue with large variations in density. Moreover, it is not flexible enough in merging sub-clusters to build main clusters in the application. Therefore, in this paper we introduce improvements to the previous solution by applying Euclidean metric instead of using SNN degree. We also compare this approach with other approaches in reducing real world large spatio-temporal datasets.

The rest of the paper is organised as follows. In Section 2 we discuss related work with different approaches for reducing spatial-temporal datasets. We résumé the different approaches for reducing spatial-temporal datasets. In section 3, we present our new hybrid clustering algorithm, which is an extension of [5], based on the

Euclidean metric distance. Section 4 presents the evaluation of this approach on very large spatio-temporal datasets of hurricane Isabel [17]. In Section 5, we discuss future work and conclude.

2 Related Work

In spite of much research in the areas of spatio-temporal data analysis and data reduction such as [6][7][10][15], there is very little literature on data reduction based on DM techniques for spatio-temporal data. Until now, to the best of our knowledge, there are only our two approaches in this paradigm that were presented in [5][11]. In these papers, the authors studied the feasibility of using DM techniques for reducing large size of spatio-temporal datasets. As mentioned in the previous section, most of the current reduction techniques do not pay attention to spatial properties of the datasets. Hence, they propose to apply DM techniques to reduce the size of the datasets without losing important spatial information by retrieving essential knowledge from these datasets. The main idea is to reduce the size of the data by producing a smaller, knowledge-oriented representation of the dataset, as opposed to compressing the data and then uncompressing it later for reuse. The main reason is to reduce and transform the data so that it can be managed and mined interactively.

In the first approach [11] the centre-based clustering technique used is K-Medoids. The k-medoids algorithm chooses the closest data object to the centre of the cluster as the cluster representative. This is very important as the technique uses the spatial and temporal attributes of the medoid points to visualise the clusters with their representatives (medoid points). This was the main advantage offered by k-medoids algorithm over other centre-based algorithms such as k-means, which would create new values for the centres based on all the members of their clusters but would have no spatial or temporal attributes. So the goal here is to find data objects where each object represents one cluster of raw data (i.e. a cluster representative). The experimental results show that the knowledge extracted from the mining process can be used as efficient representatives of huge datasets. However its representatives (medoids points) cannot reflect adequately all important features of the datasets as mentioned in section 1.

In the second approach [5], the authors have combined the density-based and the graph-based clustering in their algorithm called *snnDBS*. They have chosen a density-based method rather than other clustering method such as centre-based because it is efficient with spatial datasets as it takes into account the shape of the data objects [13]. However, there may be a performance issues when a simple density-based algorithm is applied on huge amount of spatial datasets including density variations. Indeed, the execution times as well as the choice of suitable parameters have direct impact on the complexity of density-based algorithms [2]. In this approach, a modified version of DBSCAN [13] is used because it is simple; it is also one of the most efficient density-based algorithms, applied not only in research but also in real applications. In order to cope also with the problem of density fluctuation, the authors combine DBSCAN with a graph-based clustering algorithm. Concretely, the SNN Similarity algorithm [14] is used to firstly build a similarity graph. Next, DBSCAN will be carried out based on the similarity degree. The advantage of SNN is that it

addresses the problems of low similarity and density variations. Another approach with a combination of SNN and DBSCAN was proposed in [16]. However, it is not in the context of data reduction and it did not take into account the problem of large sizes of the datasets in the context of memory constraint. A more detailed explanation on the combination of SNN and DBSCAN can be found in [5][16].

Both these approaches are incorporated in a spatio-temporal data mining framework [10][12]. This framework consists of two layers: mining and visualisation. The mining layer implements a mining process along with the data preparation and interpretation. The visualisation layer contains different visualisation tools that provide complementary functionality to visualise and interpret mined results. One of the main challenges for this framework is how to deal with the very large size of spatio-temporal datasets as they are too computationally complex for any traditional mining algorithm. Therefore in the mining layer we apply a two-pass strategy, where the goal is to reduce the size of that data by producing a smaller representation of the dataset so that it can be managed and mined efficiently. The purpose of the first pass is to group the data according to their similarity and represent these groups without losing any relevant information. Then for the second pass the objective is to apply mining technique such as clustering, association rules, etc., on the new data representatives to produce new knowledge and prepare for evaluation and interpretation.

3 Hybrid Clustering Algorithm

In this section, we propose a new metric to improve the *snnDBS* algorithm mentioned in the previous section. We analyse firstly the effect of large variety in density from the analysed datasets on the performance of this algorithm. Then, we present our new approach of density-based clustering that applies the Euclidean metric in determining the representative objects.

3.1 Issues of SNN_{degree}

There are two important remarks on the *snnDBS* algorithm. Firstly, the definition of SNN_{degree} of a point is based on the number of its shared neighbours. Secondly, the definition of a *core point* is based on the number of its neighbours that have a high SNN_{degree} . Thus, in this algorithm, the density of a point is based on the number of its' nearest neighbours (greater than $MinPts$) that have a SNN_{degree} greater than a threshold ϵ .

The issue raised from this approach is to determine the optimal threshold for the two parameters: $MinPts$ and ϵ . As shown in [18], they can be set by using heuristics and experiments. However, in the case where there is a large variation in density, it is difficult to determine a good threshold. A fixed pair of parameters ($MinPts, \epsilon$) would be an issue of quality of data reduction: if we apply a best value of parameters from a very dense area to a very low density area then we may obtain core points from the set of noise points and some clusters can be consequently created by noisy elements. The reason is that for the *snnDBS* algorithm ϵ just takes into account the number of points (nearest neighbours, shared neighbours) not the distance between them. Obviously, this issue leads to the creation of a large number of clusters. Therefore, this approach

is not good enough in merging sub-clusters [13] to build main, important clusters due to a large number of sub-clusters. Normally, different parameters which are applied to different areas of datasets would be considered as a solution such as the Optics algorithm [19]. However, the complexity of the algorithm and the execution time would cause other performance issues.

3.2 New SNN-DBSCAN Algorithm

In this section, we present a new density-based clustering algorithm *snnMDBS* which is also a combination of SNN and DBSCAN to solve the problem of density variation in the data. This algorithm uses a new density measure for the data, based on a user specified Euclidean metric radius and SNN similarity measure.

In this algorithm, the density of a point is also based on the SNN_{degree} of its' nearest neighbours. However, we propose a new definition of the nearest neighbours as well as the strategy of determining core points. In other words, we introduce a new definition for the density of a point that is based on $SNNSum$ (the sum of SNN_{degree} for all its nearest neighbours). Moreover, the Euclidean metric is applied to determine the neighbourhood of a point, i.e. the $SNNSum$ only takes into account the shared points in the neighbourhood within a radius ϵ of x . The use of the Euclidean metric for ϵ in the definition of nearest neighbours has proven to be efficient in density-based clustering algorithms [13].

Given $(MinPts, \epsilon)$ and a data point x :

- For $N_\epsilon(x)$: the neighbourhood within a radius ϵ of x ,
- For $N^k(x)$: k -nearest neighbours of x , $N^k(y)$: k -nearest neighbours of y ,
- if $x \in N^k(y)$, $y \in N^k(x)$ then $SNN_{degree}(x,y) = count(n^{(x,y)})$, $n^{(x,y)} \in N^k(x)$, $n^{(x,y)} \in N^k(y)$; else $SNN_{degree}(x,y) = 0$.

$$- SNNSum(x) = \sum_{i=1}^n SNN_{degree}(x, x'_i), \forall x'_i \in N_\epsilon(x).$$

x is a core point if:

- $SNNSum(x) \geq MinPts$,

There are two important differences in the definition of core points between the *snnMDBS* and *snnDBS* algorithm [5]. Firstly, the $SNNSum$ only takes into account the shared neighbours in the set of x 's neighbourhood with respect to the radius ϵ . This metric is used to reduce the large variation of densities for each point compared to the previous definition. Consequently, we can decrease the number of core points which are classified as noise objects. Fig. 1 shows how this new density measure performs on the point P1 with respect to its k -nearest neighbours ($k=15$). The number assigned to each neighbour is the SNN_{degree} between the neighbours and P1. The $SNNSum$ is calculated by adding the SNN_{degree} of each neighbour inside the radius ϵ . P1 is then classified as a core point if $SNNSum(x) \geq MinPts$.

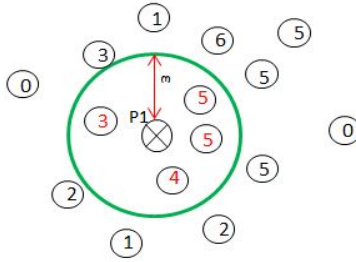


Fig. 1. Density measure based on distance metric and SNN

Using the *snnDBS* approach from [5], it has proven to be difficult to form clusters with points of different densities, as a consequence the resulting number of clusters was always very high. Secondly, in the case of *snnMDBS*, the last condition in the definition of a core point x ($\text{SNNSum}(x) \geq \text{MinPts}$) is stronger than the *snnDBS* algorithm in terms of the density and of the relationship with its neighbours because it takes into account not only the number of its neighbours but also their SNNSum (wrt. the ϵ). In the next section we will show the performance of this new approach as well as comparing it with existing approaches.

4 Evaluation and Analysis

In this section we evaluate firstly our new approach *snnMDBS* under different criteria. Next, we compare it with the approach *snnDBS* presented in [5] and then with the DBSCAN algorithm [14]. We apply four criteria to evaluate this new approach: (1) the ratio (called compression ratio) of data size before and after applying the algorithm; (2) finding the optimal parameters for the algorithm; (3) the number of clusters each algorithm produces; (4) the shape of the clusters produced by each algorithm. The reason for choosing these criteria has been discussed in [18].

4.1 Experiment Setup

The dataset is the Isabel hurricane data [17] produced by the US National Centre for Atmospheric Research (NCAR). It covers a period of 48 hours (time-steps). Each time-step contains several atmospheric variables. The grid resolution is $500 \times 500 \times 100$. The total size of all files is more than 60GB (~ 1.25 GB for each time-step). Datasets of each time-step include 13 non-spatial attributes, so-called dimensions. In this evaluation, the dimensions QCLOUD (cloud moisture mixing ratio) and QICE (cloud ice mixing ratio) are chosen for analysis; the weight of the cloud water and ice measured at each point of the grid. The range of QCLOUD value is $[0 \dots 0.00332]$ and for QICE value is $[0 \dots 0.00054168125]$. Totally, the testing datasets contains around 25 million data points of four dimensions X, Y, Z and QCLOUD or QICE for each time step. The evaluation is carried out on six different time-steps: 2, 10, 18, 26, 34 and 42. We also filter the NULL values and land values in the testing datasets.

4.2 Analysis

As shown in Table 1, the compression ratio is always around 1:10 of the whole datasets. This is very important in the case of analysing very large datasets. If it can preserve the important information this would mean that users would only need to analyse 10% of the whole datasets.

The next issue that needs to be dealt with is the optimal values for the two parameters $(MinPts, \epsilon)$ of the algorithm *snnMDBS*. As this algorithm is a DBSCAN-based algorithm, we can apply a similar approach proposed in [13] to determine the initial values for these two parameters. The difference is that we arrange data according to their *snnSum* rather than their *k-distance* and select an ϵ value that minimises the noise. We carried out brief tests with original DBSCAN on a testing set consisting of values for QCLOUD in time step 2. Taking into consideration the fact that we normalise the data values for DBSCAN we found that values between 0.01 and 0.02 are good candidates for ϵ and that a value of 8 was a good candidate for the minimum *SNNSum* (*MinPts*) a dense point can have within its ϵ -radius.

Next, we test these candidates for different time-steps of QCLOUD to determine the optimal pair of $(MinPts, \epsilon)$. The factors that affected this decision were the minimum number of noise points so that very little information would be lost, the maximum number of core points so that the shape of the data will be as close as possible to the original and the minimum number of clusters so that the maximum reduction rate can be achieved.

Table 1 shows the results of the evaluation of *snnMDBS* algorithm for different time-steps of QCLOUD with $(MinPts, \epsilon)$ equal to (8, 0.01). The number of noise points is negligible compared to the dataset size (only 0.0007% of the whole datasets) as well as the number of the representatives (only 0.006%). We obtain moreover appropriate number of clusters. For example, in the case of time-step 2, the number of clusters is 115 of 946569 data points. This means that one cluster could have around 8000 points (0.8% of the data).

Table 1. Results for *snnMDBS* on different QCLOUD time steps

Time step	MinPts	ϵ	Dataset size	Noise	Representatives	Clusters	<i>snnDBS</i> - Clusters ¹
2	8	0.01	946569	34	103898	115	1552
10	8	0.01	809244	728	87770	222	3470
18	8	0.01	775150	846	82067	351	3683
26	8	0.01	967519	1104	103087	425	4178
34	8	0.01	1134109	1157	121266	356	4816
42	4	0.01	1243338	1258	133247	447	4701

¹ Number of clusters created by the algorithm *snnDBS* with its optimal parameters[18].

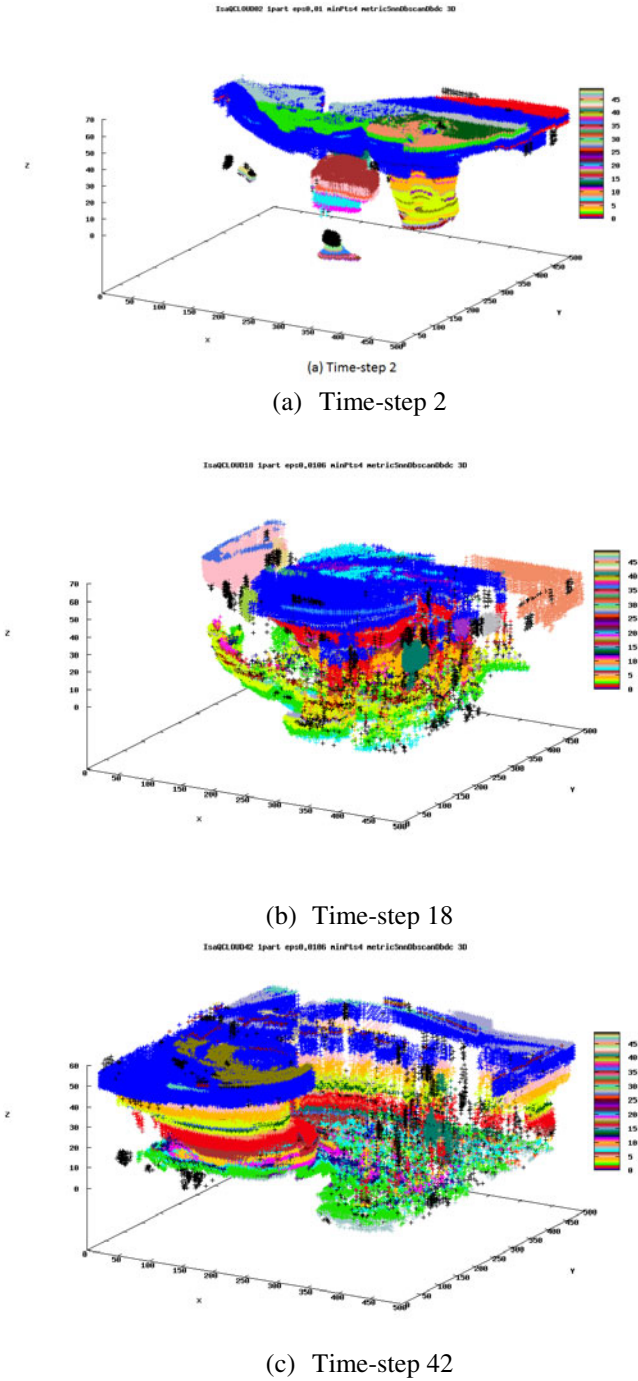


Fig. 2. 50 largest clusters produced by *smnMDBS* on QCLOUD

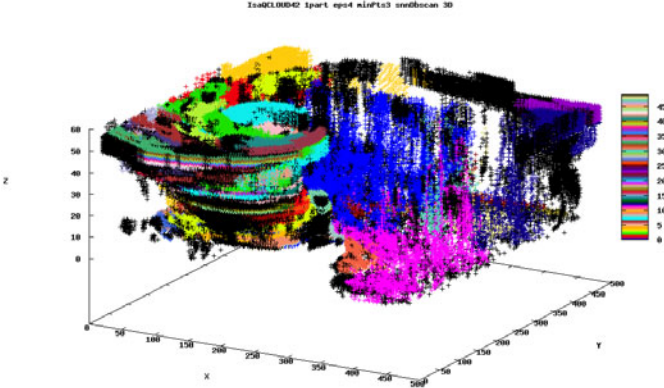
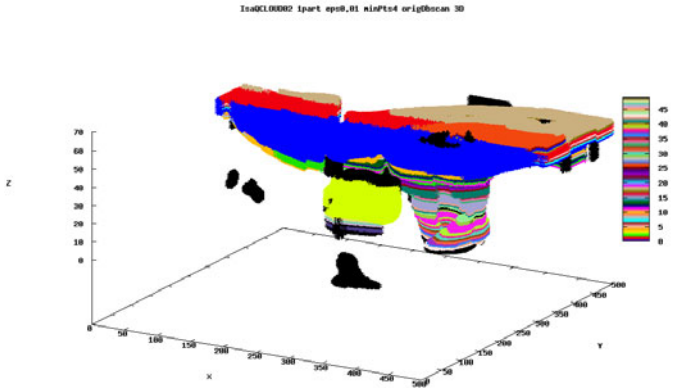


Fig. 3. 50 largest clusters produced by *snnDBS* on QCLOUD for time step 42

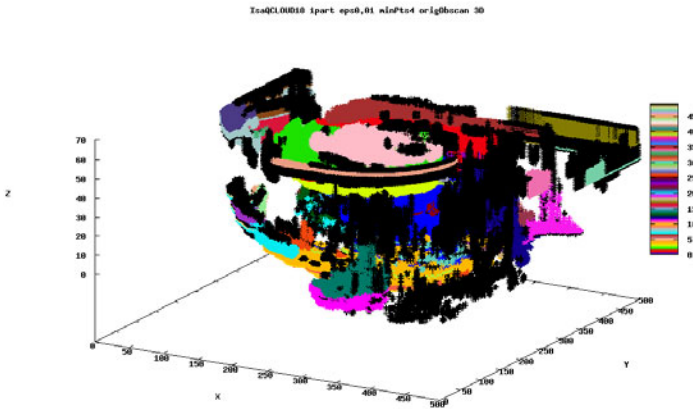
Furthermore, Table 1 shows the comparison on the number of clusters created by the two algorithms: *snnMDBS* and *snnDBS*. Obviously, we have an important improvement in terms of reducing the number of clusters (up to 93%). As discussed in [18], a high number of clusters is a performance issue. The main reason for this poor performance by *snnDBS* compared with *snnMDBS* is the rigidity of its ϵ parameter based on the SNN_{degree} to merge sub-clusters. A good density measure of a core point by introducing the Euclidean metric for ϵ is important. This helped to limit the effect of high variation of densities in the dataset. Consequently, *snnMDBS* could form a stronger set of clusters resulting in smaller number of clusters.

Fig. 2 shows the 3D views of the 50 largest clusters for three of the chosen QCLOUD time step (2, 18, 42) datasets produced by the *snnMDBS* algorithm with the parameter (MinPts, ϵ) taken from Table 1. Fig. 3 shows the 50 largest clusters produced by the *snnDBS* algorithm on QCLOUD time step 42 with its best parameter values [18]. In the graphs of both Fig. 2 and 3 the shape and movement of the hurricane is clearly visible. This is indicated by the distinct swirling shape in each of the graphs. The clear difference in the results of both algorithms is the clusters of the dense regions that indicate the structure of the hurricane and the size of cluster 49 represented in black. This colour represents all clusters with the exception of the 49 largest clusters. The graphs of *snnDBS* in Fig. 3 show a larger size for the black cluster 49 which contains the main structure of the hurricane. In Fig. 2 this is not the case, the cluster in black is insignificant in each of the graphs. The reason for this is that *snnDBS* is not efficient enough in merging sub-clusters, due to the lack of a good distance measure in determining its representatives.

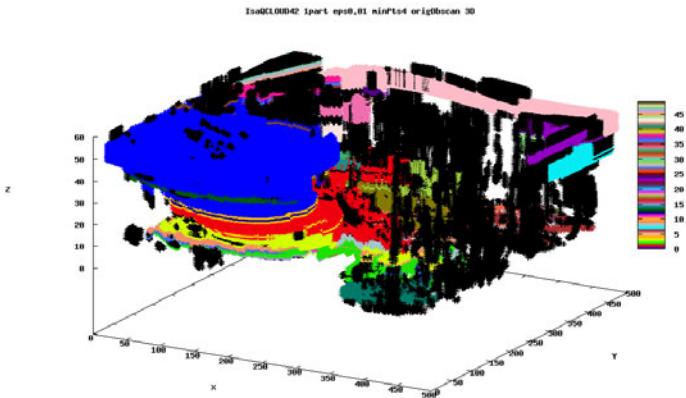
In order to evaluate the quality of the visual shape of the clusters produced by this *snnMDBS* algorithm, we also implemented the DBSCAN algorithm [13]. Fig. 4 is the 3-dimensional views of the 50 largest clusters for three of the QCLOUD time step (2, 18, 42) datasets produced by the DBSCAN algorithm. By observing this figure as well as comparing it to the Fig. 2, we recognize that the *snnMDBS* algorithm can preserve important information of QCLOUD. In other words, it preserves most of the important visual shapes comparing to the DBSCAN.



(a) Time-step 2



(b) Time-step 18



(c) Time-step 42

Fig. 4. 50 largest clusters produced by DBSCAN on QCLOUD

Table 2 presents the results of *snnMDBS* on QICE. The data analysis for QICE is very different to that for QCLOUD as the size of QICE data is huge (5,224,696 for compared to 946,569 for QCLOUD). The similarity matrix used for calculating the k-nearest neighbours is too large to fit in memory, so a spatial index structure (kd tree) was implemented to produce the similarity matrix more efficiently. It is clear that there is a similar improvement in the quality of results for *snnMDBS* on QICE as presented for QCLOUD in Table 1. The number of noise points and representatives are negligible compared to the overall dataset size and the number of clusters is much small than what was produced by DBSCAN.

Fig. 5 shows the 3D view of the 50 largest clusters for QICE time step 18 and 42 datasets produced by the *snnMDBS* algorithm with the parameters (minPts , ϵ) taken from Table 2. To evaluate the visual shape of Fig. 5 we compare it to the results of DBSCAN on the same data. The results are shown in Fig. 6. The visual shape of the resulting clusters for both algorithms is very similar, thus *snnMDBS* can preserves the same important visual information compared to DBSCAN.

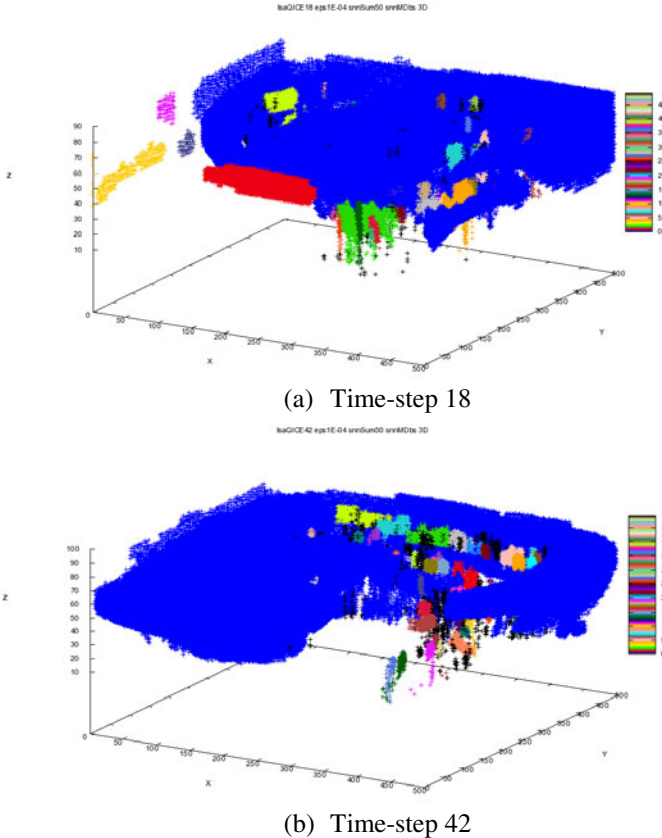
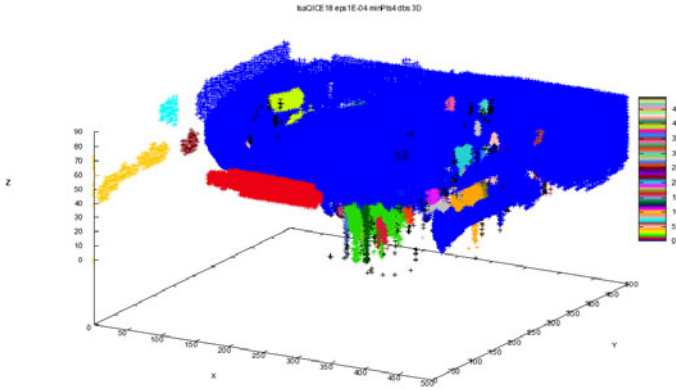
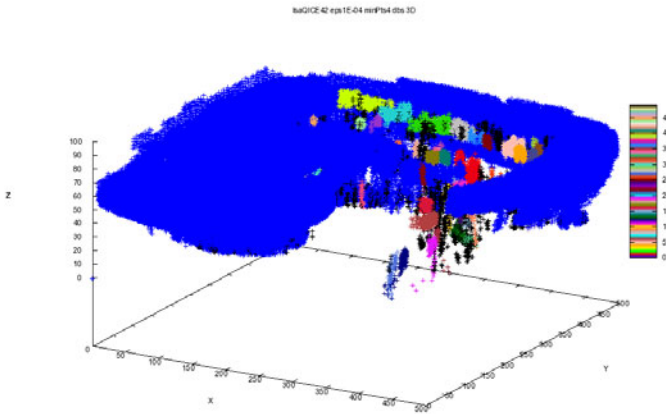


Fig. 5. 50 largest clusters produced by *snnMDBS* on QICE



(a) Time-step 18



(b) Time-step 42

Fig. 6. 50 largest clusters produced by DBSCAN on QICE**Table 2.** Results for *snnMDBS* on different QICE time steps

Time step	MinPts	ϵ	Dataset size	Noise	Representatives	Clusters	DBS-Clusters ²
2	50	0.0001	5224696	2823	721714	3	128
10	50	0.0001	3244443	7199	420092	113	624
18	50	0.0001	2195180	7789	278203	161	724
26	50	0.0001	2278961	8338	287994	189	777
34	50	0.0001	2523664	11050	318426	184	1050
42	50	0.0001	2746494	13115	349181	256	1381

² Number of clusters created by the algorithm DBSCAN with minPts value 4 and ϵ value 0.0001 as its parameters.

We note from Table 2 nearly all data objects (approximately 90%) are gathered in three biggest clusters in time step 2. It shows that there is a slight difference of the ratio of cloud ice mixing in the early hours of the hurricane, but as time changes and the hurricane develops this ratio is dispersed. This is proved by the fact that as time changes the number of clusters increases. This would indicate that the density of QICE values is very high at the beginning, but decreases over time. Also we note that, compared to *DBSCAN*, there is a large reduction in the number of clusters produced by *snnMDBS* (an average ratio of 8:1). The reason for this is the use of a minimum *SNNSum* instead of the traditional *minPts* as a parameter to regulate the density strength of the clusters. From our experimentations it has been noted that a high value (>50) results in a larger number of noise and a low value (<50) results in less noise but a significant increase in the number of clusters. Consequently *DBSCAN* without this notion of minimum *SNNSum* produces a large number of small fragmented clusters compared to *snnMDBS*.

5 Conclusion and Future Work

In this paper, we presented an improvement of the algorithm described in [5] to reduce very large spatio-temporal dataset. This improvement was based on the addition of a Euclidean metric distance radius to determine the nearest neighbour similarity for each core point rather than just using the number of nearest neighbours of the core points. We compared the performance of this new approach against both the previous one [5] and *DBSCAN* algorithm [13] on a real-world spatio-temporal datasets. The experimental results show that the new definition for the density of a point greatly improves the clustering results of the *SNN_DBSCAN*-based algorithm. Besides, it also shows that knowledge extracted can be used as efficient representatives of huge datasets. Furthermore, we do not lose any important information from the data that could have an adverse effect on the result obtained from mining the data at a later stage.

In the future we intend to provide a more extensive evaluation involving the analysis of more dimensions other than *QCLOUD* and *QICE*. Furthermore parallel and distributed techniques will also be studied to carry out our approach on both multi-core and distributed architecture in order to prove its.

References

1. Dunham, M.H.: *Data Mining: Introductory and Advanced Topics*. Prentice Hall (2003)
2. Tan, P.-N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*. Addison Wesley (2006)
3. Ye, N. (ed.): *The Handbook of Data Mining*. Lawrence Erlbaum Associates Publishers, Mahwah (2003)
4. Johnston, W.L.: Model visualisation. In: *Information Visualisation in Data Mining and Knowledge Discovery*, pp. 223–227. Morgan Kaufmann, Los Altos (2001)

5. Le-Khac, N.-A., Bue, M., Whelan, M., Kechadi, M.-T.: A Clustering-Based Data Reduction for Very Large Spatio-Temporal Datasets. In: Cao, L., Zhong, J., Feng, Y. (eds.) ADMA 2010, Part II. LNCS, vol. 6441, pp. 43–54. Springer, Heidelberg (2010)
6. Roddick, J.F., Hornsby, K., Spiliopoulou, M.: An updated bibliography of temporal, spatial, and spatio-temporal data mining research. In: Roddick, J., Hornsby, K.S. (eds.) TSDM 2000. LNCS (LNAI), vol. 2007, pp. 147–163. Springer, Heidelberg (2001)
7. Roddick, J.F., Lees, B.G.: Paradigms for Spatial and Spatio-Temporal Data Mining. In: Miller, H., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery. Taylor & Francis (2001)
8. Kivinen, J., Mannila, H.: The power of sampling in knowledge discovery. In: Proceedings of the ACM SIGACT-SIGMOD-SIGART, Minneapolis, Minnesota, United States, May 24-27, pp. 77–85 (1994)
9. Sayood, K.: Introduction to Data Compression, 2nd edn. Morgan Kaufmann (2000)
10. Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., Kechadi, T.: Exploratory Spatio-Temporal Data Mining and Visualization. *Journal of Visual Languages and Computing* 18(3), 255–279 (2007)
11. Whelan, M., Le-Khac, N.-A., Kechadi, M.-T.: Data Reduction in Very Large Spatio-Temporal Data Sets. In: IEEE International Workshop On Cooperative Knowledge Discovery and Data Mining 2010 (WETICE 2010), Larissa, Greece (June 2010)
12. Bertolotto, M., Di Martino, S., Ferrucci, F., Kechadi, T.: Towards a Framework for Mining and Analysing Spatio-Temporal Datasets. *International Journal of Geographical Information Science* 21(8), 895–906 (2007)
13. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering clusters in Large Spatial Databases with Noise. In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD 1996), Portland, OR, USA, pp. 226–231 (1996)
14. Januzaj, E., Kriegel, H.-P., Pfeifle, M.: DBDC: Density-Based Distributed Clustering. In: Jarke, M., Bubenko, J., Jeffery, K. (eds.) EDBT 1994. LNCS, vol. 779, pp. 88–105. Springer, Heidelberg (1994)
15. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared nearest neighbours. *IEEE Transactions on Computers* C-22(11), 1025–1034 (1973)
16. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of Second SIAM International Conference on Data Mining (2003)
17. National Hurricane Center, Tropical Cyclone Report: Hurricane Isabel (2003), <http://www.tpc.ncep.noaa.gov/2003isabel.shtml>
18. Le Khac, N.-A., Whelan, M., Kechadi, M.-T.: Performance Evaluation of a Density-based Clustering Method for Reducing Very Large Spatio-temporal Dataset. In: The 2011 International Conference on Information and Knowledge Engineering (IKE 2011), Las Vegas, USA, July 18-21 (2011)
19. Ankerst, M., Breunig, M., Kriegel, H.-P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: ACM SIGMOD International Conference on Management of Data, pp. 49–60. ACM Press (1999)

A Normal Distribution-Based Over-Sampling Approach to Imbalanced Data Classification

Huaxiang Zhang^{1,2} and Zhichao Wang^{1,2}

¹ Department of Computer Science, Shandong Normal University,
Jinan 250014, Shandong, China

² Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology,
Jinan, 250014, China
huaxzhang@hotmail.com

Abstract. This study proposes a normal distribution-based over-sampling approach to balance the number of instances belonging to different classes in a data set. The balanced training data are used to learn unbiased classifiers for the original data set. Under some conditions, the proposed over-sampling approach generates samples with expected mean and variance similar to that of the original minority class data. As the approach tries to generate synthetic data with similar probability distributions to the original data, and expands the class boundaries of the minority class, it may increase the minority class classification performance. Experimental results show that the proposed approach outperforms alternative methods on benchmark data sets most of the times when implementing several classical classification algorithms.

Keywords: imbalanced classification, over-sampling, normal distribution.

1 Introduction

Imbalance data sets [1] refer to those that the number of one class instances is far more than that of another class instances for each concept in the training data. It is quite common in practice and automatic concept learning from imbalanced data sets usually produces biased classifiers with high predictive accuracy on the majority class data, but poor predictive accuracy on the minority class data [2].

Learning from imbalanced datasets has been explored extensively, and the existing works can be divided into data and algorithmic levels. The algorithm-level approaches use the original training data to construct new algorithms suitable to imbalanced data sets, and data-level approaches[3-6] generate new training datasets from the original datasets to make their class distribution approach balanced. As under-sampling may discard potentially useful data that could be important for classification, and over-sampling may change the original minority class sampling distribution.

Over-sampling refers to the process of generating more training instances for the minority class to balance the class distribution. As research results show that over-sampling with replication does not significantly improve the classification accuracy of the minority class data. A novel approach SMOTE [7] is proposed to

overcome this issue. SMOTE is a linear over-sampling method which adopts linear interpolations between two near samples to generate synthetic samples. SMOTE ignores the change of the underlying probability distribution of the minority class data after synthetic samples have been included in the training data. Borderline-SMOTE methods [8] in which only the minority samples near the borderline are over-sampled. Borderline-SMOTE produces little change in performance and sometimes hurts the generalization of an algorithm, as it changes the minority data distribution. Data-Boost-IM [9] combined synthetic data generation and an ensemble learning algorithm to tackle the imbalanced data classification problem. Random Over-Sampling (RO-Sampling) is a non-heuristic method that aims to balance class distribution through randomly duplicating the minority class instances. After the minority class instances have been processed, there will be several exact copies of some minority class instances in the new training data set, and this increases the likelihood of occurring over-fitting. Combination of over-sampling and under-sampling is often performed to resolve the imbalance problems [10] [11].

It is reported that under-sampling produces a reasonable sensitivity to changes in class distribution, and over-sampling often produces little or no change in performance as the training data distribution has been changed [12]. We just focus on the over-sampling approaches to handling imbalanced data classification, and ignore attribute processing methods [13] [14] in this work.

All the above mentioned data processing methods, whether they perform over-sampling, under-sampling or combinations of the both on the original datasets, will result in changing the sampling distribution of the original data, thus leads to a biased classifier that is not quite suitable to classify the original data.

We propose a novel over-sampling approach called normal distribution-based over-sampling (NDO-Sampling). To our knowledge, it differs from all the proposed over-sampling approaches in the related literature, such as non-heuristic over-sampling and heuristic over-sampling, and it neither performs linear interpolations between neighbors nor increases the number of minority class data with replacement, it generate synthetic samples through implementing a random walk starting from the original data point. Under some assumptions and when some conditions are satisfied, the expected value and the squared deviation of the synthetic samples can be proved to be probably approximate the same as that of the original minority class data. Our method decreases the risk of increasing the likelihood among instances in the minority class.

We organize the paper as follows. Section 2 describes the evaluation metrics commonly used in imbalanced data classification. Section 3 presents the normal distribution-based over-sampling model, and proves two theorems concerning the mean and the squared deviation of the synthetic samples. Section 4 evaluates NDO-Sampling on broad data sets from different aspects. Section 5 concludes the paper.

2 Performance Evaluation Metrics

The overall accuracy on a test dataset is commonly used to evaluate the performance of a classifier. But for imbalanced data, as the accuracy is profoundly dominated by the minority class data, alternative evaluation metrics are employed. These appropriate

metrics include Area Under the ROC Curve (*AUC*), *F-Measure*, Geometric Mean (*G-Mean*), *overall accuracy* and the accuracy rate for the minority class. We refer minority class to positive class and majority class to negative class. After the four values *TP*(the number of true positive), *FP*(the number of false positive), *TN*(the number of true negative) and *FN*(the number of false negative) have obtained, *precision* and *recall* are calculated as $TP/(TP+FP)$ and $TP/(TP+FN)$ respectively, positive accuracy(*pa*) and negative accuracy(*na*) are calculated as $TP/(TP+FN)$ and $TN/(TN+FP)$ respectively, and then *F-Measure* and *G-Mean* are defined as

$$F\text{-Measure} = \frac{(1 + \beta^2) \times \text{recall} \times \text{precision}}{(\beta^2 \times \text{recall} + \text{precision})} . G\text{-Mean} = \sqrt{\text{pa} \times \text{na}} .$$

We set β to 1 in this paper.

The above metrics represent different aspects of the learning algorithms, and are extensively used in the field of imbalanced data classification problems. *F-measure* [15] integrates *precision* and *recall* into a single metric, and measures how “good” a learning algorithm on the interested class. It is high when both the *recall* and *precision* are high. The ROC curve indicates a balanced classification ability of a classifier by considering the tradeoffs between *TP Rate* and *FP Rate* [16]. *G-Mean* [17] measures whether the accuracy on each of the two classes are maximized.

We use *F-measures* for both majority class and minority class, the *overall accuracy*, *G-mean* and *TP rate* as the evaluation metrics in this work.

3 The Normal Distribution Model

The central limit theorem states that no matter what the real sampling distribution is, the sampling distribution of the mean approaches a normal distribution. This inspires us to create synthetic samples for the minority class without knowing the real sampling distribution. We aim to generate samples approximately obeying the real sampling distribution. After the newly generated samples are put together with the original minority class instances, these newly formed training data will keep the original sampling distribution approximately unchanged.

We first make the following attribute independence assumption: each attribute of the training data is considered as a random variable, and all the attributes are independent of each other. Given the m attributes, denoted as a_1, a_2, \dots, a_m , we have m random variables. Based on the given minority class data, we calculate the expected value and variance for each random variable, and the mean and the standard variance of a_i are denoted as μ_i and σ_i respectively, where $i \in \{1, 2, \dots, m\}$.

Let μ_i' denote the mean of the unknown underlying distribution governing random variable a_i and σ_i' be the standard deviation. We say that all the values of attribute a_i for the minority training data are independent, identically distributed random variable values, because they represent independent experiments, and each obeying the similar underlying probability distribution. According the conclusion of central limit theorem,

we know that, as the number n of samples approaches infinite, the distribution governing the following random variable approaches a Normal distribution, with zero mean and standard deviation equals 1.

$$\frac{\mu_i - \mu_i'}{\sigma_i' / \sqrt{n}} \xrightarrow{P} N(0, 1) . \quad (1)$$

n is the number of minority class instances. Inspired by (1), given the value r_i of a random variable obeying distribution $N(0, 1)$, we have the following equation:

$$\mu_i' = \mu_i - r_i \bullet \sigma_i' / \sqrt{n} . \quad (2)$$

In (2), μ_i is the mean of attribute a_i for the given training minority class data, and we consider it as the representative of the original minority class data. μ_i' is the mean of attribute a_i for the unknown minority class data, and we consider it as the representative of the unknown minority class data. So for any instance and its given value of a_i , we can generate a synthetic value for this attribute through the following calculation.

$$a_i' = a_i - r_i \bullet \sigma_i' / \sqrt{n} , \quad i \in \{1, 2, \dots, m\} . \quad (3)$$

In (3), a_i' is a new value of attribute a_i . σ_i' is unknown, we use σ_i to approximate it, and obtain equation (4)

$$a_i' = a_i - r_i \bullet \sigma_i / \sqrt{n} , \quad i \in \{1, 2, \dots, m\} . \quad (4)$$

We call (4) a normal distribution model. Based on this model, we propose a normal distribution-based over-sampling approach.

We obtain two conclusions from (4).

Theorem 1. *As n approaches infinite, the expected mean of the random variable values of attribute a_i obtained using (4) equals μ_i .*

Theorem 2. *The expected standard deviation of the random variable values of attribute a_i obtained using (4) equals σ_i as n approaches infinite.*

The conclusions of the above two theorems (duo to the limit to the range of pages, the related proofs were not shown in this paper) tell us, if we generate a synthetic value for each attribute according to (3), we get m synthetic values, and the m values form a vector, which can be considered as a synthetic training instance. Given the over-sampling rate, we create required number of instances for the minority class, and construct a classifier using the minority class training data together with the synthetic data. As the expected mean and the expected squared deviation of the synthetic samples equal the mean and the standard deviation of the original minority class data correspondingly, the constructed classifier will be suitable to the training data well and will be an unbiased classifier.

4 Experimental Results

All the experiments are conducted on 8 benchmark datasets extensively used in classification tasks. These datasets are selected from the UCI Machine Learning Repository¹ and are summarized in table 1. They have different data sizes and degrees of skew, and come from different domains, thus making the experimental results much more convincing. We do several sets of experiments in order to evaluate NDO-Sampling from different aspects. We use *TP rate*, *F-measure*, *G-mean* and *overall accuracy* to evaluate the results of our experiments, and study the impacts of varying imbalance ratios. In the first set of experiments, we employ three baseline classifier algorithms to compare NDO-Sampling with SMOTE and RO-Sampling. These algorithms are C4.5, NB (Naive Bayes) and KNN($k=3$). Each experiment is done 10 times independently for each baseline algorithm, and in each time, a ten-fold cross validation is applied. The end results for each algorithm are the average of the 10 independent ten-fold cross validation experiments. In the second set of experiments, we study the impacts of different over-sampling approaches on multi-class classification problems.

The minority class is over-sampled at 100%, 200% and 300% of its original size, and the number of nearest neighbors is set to 5 for SMOTE. For the Glass, Satimage, Segment-challenge and Vehicle, we choose the smallest class as the positive class and convert the rest of the classes into a single class as the negative class to increase the degree of skew. The datasets are described in table one.

Table 1. Summary of datasets (including dataset, number of instances, number of minority class instances, number of attributes and minority class label)

dataset	# of inst.	# of min. inst.(%)	# of feat.	target
Breast-w	699	241(34.48)	9	class=malignant
Diabetes	768	268(34.90)	8	class=tested positive
Glass	214	9(4.21)	9	type=tableware
Ionosphere	351	126(35.90)	34	class=b
Satimage	6430	625(9.72)	36	class=2.
Segment-challenge	1500	205(13.67)	19	class=brick face
Sonar	208	97(46.63)	60	class=Rock
Vehicle	846	199(23.52)	18	class=van

We use Weka² to implement the experiments. Results for the above data sets are shown in table 2. Different baseline algorithms are executed at different over-sampling rates. We use F-min to denote *F-measure* for the minority class, F-maj to denote *F-measure* for the majority class, O-acc. to denote the *overall accuracy*, Alg. to denote algorithms, NDO, SMO and RO to denote the over-sampling strategies normal distribution over-sampling, SMOTE and random over-sampling respectively, and O.S to denote Over-Sampling strategies in the following tables and figures.

¹ <http://www.ics.uci.edu/~mllearn/MLRepository.html>

² <http://www.cs.waikato.ac.nz/ml/weka>

Table 2. (1). Results on 8 datasets with continuous attributes
(over-sampling rate at 100% and 200%)

dataset	Alg.	O.S	100%					200%				
			F-min (%)	F-maj (%)	O-acc (%)	G-mean (%)	TPrate (%)	F-min (%)	F-maj (%)	O-acc (%)	G-mean (%)	TPrate (%)
Breast-w	C4.5	NDO	93.12	96.24	95.14	95.23	95.52	92.95	96.10	94.98	95.22	96.02
		SMO	92.80	96.11	94.95	94.83	94.47	92.71	96.02	94.85	94.89	95.02
		RO	92.39	95.94	94.71	94.35	93.22	92.14	95.73	94.47	94.37	94.05
	NB	NDO	94.38	96.89	95.99	96.35	97.51	94.38	96.89	95.99	96.35	97.51
		SMO	94.38	96.89	95.99	96.35	97.51	94.38	96.89	95.99	96.35	97.51
		RO	94.38	96.89	95.99	96.35	97.51	94.38	96.89	95.99	96.35	97.51
	3-NN	NDO	95.94	97.77	97.12	97.45	98.51	96.39	98.00	97.42	97.96	99.75
		SMO	96.36	98.01	97.42	97.73	98.76	96.50	98.08	97.52	97.90	99.17
		RO	95.56	97.62	96.90	96.88	96.82	95.21	97.38	96.61	96.82	97.51
Diabetes	C4.5	NDO	67.31	78.52	74.08	74.61	76.49	66.43	77.04	72.73	73.71	77.31
		SMO	66.30	77.78	73.22	73.73	75.50	64.84	74.50	70.44	71.98	78.11
		RO	60.52	76.88	70.83	69.06	64.05	60.18	75.66	69.79	68.70	65.42
	NB	NDO	65.93	79.73	74.58	73.57	70.49	66.74	77.78	73.36	74.07	76.60
		SMO	66.27	80.25	75.09	73.84	70.15	65.59	76.71	72.22	73.01	75.87
		RO	65.57	79.95	74.65	73.26	69.15	65.35	77.36	72.61	72.93	74.00
	3-NN	NDO	65.03	75.61	71.26	72.38	76.57	74.42	78.38	76.56	79.83	97.69
		SMO	65.14	76.12	71.66	72.56	75.87	65.73	73.68	70.23	72.37	81.84
		RO	60.07	74.21	68.66	68.40	67.54	60.54	69.69	65.71	67.55	75.37
Glass	C4.5	NDO	94.74	99.76	99.53	99.76	100.00	94.74	99.76	99.53	99.76	100.00
		SMO	94.74	99.76	99.53	99.76	100.00	94.74	99.76	99.53	99.76	100.00
		RO	94.74	99.76	99.53	99.76	100.00	94.74	99.76	99.53	99.76	100.00
	NB	NDO	68.44	97.93	96.12	97.95	100.00	66.91	97.78	95.84	97.81	100.00
		SMO	69.44	98.18	96.57	94.65	92.59	71.64	98.44	97.04	93.05	88.89
		RO	64.86	97.85	95.95	92.50	88.89	60.76	97.43	95.17	92.11	88.89
	3-NN	NDO	73.30	98.55	97.24	93.70	90.00	74.69	98.49	97.15	98.50	100.00
		SMO	73.85	98.61	97.35	93.20	88.89	69.57	98.27	96.73	92.89	88.89
		RO	71.64	98.44	97.04	93.05	88.89	62.16	97.69	95.64	90.48	85.19

Table 2. (1). (Continued)

Ionosphere	C4.5	NDO	86.71	92.72	90.60	89.38	85.48	86.40	92.82	90.60	88.78	83.17
		SMO	85.45	91.73	89.46	88.72	86.24	86.43	91.76	89.74	90.02	91.01
		RO	87.60	93.26	91.26	90.01	85.98	83.99	90.92	88.41	87.54	84.66
	NB	NDO	82.34	89.62	86.92	86.47	84.92	82.63	89.84	87.18	86.66	84.92
		SMO	80.54	87.79	85.00	85.32	86.51	81.44	88.56	85.85	85.99	86.51
		RO	77.09	84.46	81.48	82.54	86.77	76.74	83.95	81.01	82.25	87.30
	3-NN	NDO	85.79	92.78	90.43	87.90	80.48	95.57	97.40	96.72	97.10	98.49
		SMO	90.61	94.97	93.45	92.18	88.10	95.02	97.05	96.30	96.75	98.41
		RO	82.11	91.43	88.41	84.52	74.07	84.00	92.03	89.36	86.34	77.78
Satimage	C4.5	NDO	63.38	95.56	92.08	81.61	70.56	68.70	96.05	92.99	86.50	79.20
		SMO	63.39	95.64	92.20	81.07	69.44	63.91	95.58	92.13	82.23	71.68
		RO	55.57	95.18	91.30	72.98	56.00	57.25	95.25	91.45	74.77	58.88
	NB	NDO	47.98	88.78	81.54	84.18	87.60	47.59	88.57	81.23	84.03	87.68
		SMO	48.87	89.40	82.44	84.15	86.35	48.76	89.32	82.33	84.16	86.51
		RO	47.96	88.81	81.58	84.10	87.36	47.52	88.50	81.13	84.06	87.89
	3-NN	NDO	68.90	95.74	92.50	89.27	85.44	74.67	96.21	93.41	96.28	100.00
		SMO	72.17	96.08	93.12	92.52	91.79	71.40	95.58	92.34	94.98	98.40
		RO	67.18	95.75	92.47	86.26	79.25	65.55	95.08	91.39	88.12	84.27
Segment-challenge	C4.5	NDO	97.23	99.56	99.24	98.59	97.71	97.82	99.65	99.40	98.97	98.39
		SMO	96.92	99.51	99.16	98.27	97.07	97.96	99.68	99.44	98.65	97.56
		RO	96.86	99.50	99.13	98.61	97.89	96.84	99.50	99.13	98.26	97.07
	NB	NDO	65.47	91.79	86.73	89.55	98.37	66.55	92.22	87.38	89.23	99.02
		SMO	62.44	89.69	83.82	88.91	92.03	61.90	89.41	83.42	89.36	98.54
		RO	61.24	89.08	82.96	89.06	98.04	61.67	89.22	83.18	89.38	98.87
	3-NN	NDO	97.91	99.66	99.42	99.35	99.24	97.82	99.65	99.39	99.62	99.95
		SMO	97.44	99.59	99.29	99.11	98.86	97.53	99.60	99.31	99.33	99.35
		RO	96.90	99.50	99.13	99.16	99.19	96.54	99.43	99.02	99.30	99.67
Sonar	C4.5	NDO	82.90	80.62	81.83	81.77	94.43	84.40	82.07	83.32	83.21	96.80
		SMO	78.15	77.94	78.04	78.22	84.19	81.75	80.72	81.25	81.39	90.03
		RO	70.63	76.48	73.88	73.21	67.35	72.70	76.67	74.84	74.60	71.82

Table 2. (1). (Continued)

	NB	NDO	71.85	63.77	68.32	67.31	86.70	73.90	65.13	70.14	68.81	90.62
		SMO	70.40	64.67	67.79	67.37	82.13	70.41	65.03	67.95	67.59	81.79
		RO	70.86	64.78	68.11	67.60	83.16	71.14	64.77	68.27	67.70	83.85
	3-NN	NDO	85.96	85.96	85.96	86.16	92.16	89.23	88.43	88.85	88.98	99.07
		SMO	88.31	88.61	88.46	88.65	93.47	87.66	87.34	87.50	87.69	95.19
		RO	87.31	87.99	87.66	87.82	91.07	86.54	86.54	86.54	86.74	92.78
Vehicle	C4.5	NDO	88.47	96.25	94.34	93.64	92.36	87.17	95.70	93.56	93.39	93.07
		SMO	86.80	95.72	93.54	92.39	90.28	86.40	95.56	93.30	92.30	90.45
		RO	86.76	95.79	93.62	91.95	88.94	86.96	95.91	93.77	91.80	88.27
	NB	NDO	55.66	72.06	65.72	72.71	91.46	56.63	71.98	65.96	73.50	94.47
		SMO	56.26	72.35	66.12	73.27	92.63	56.27	71.87	65.76	73.18	93.63
		RO	54.52	71.93	65.29	71.72	88.44	55.20	71.82	65.41	72.28	90.62
	3-NN	NDO	86.40	95.26	92.97	93.64	94.92	88.36	95.78	93.80	95.86	100.00
		SMO	87.00	95.56	93.38	93.64	94.14	86.97	95.42	93.22	94.22	96.15
		RO	85.29	94.98	92.51	92.44	92.29	84.48	94.43	91.80	92.82	94.81

Table 2 (2). Results on 8 datasets with numerical attributes(over-sampling rate at 300%)

dataset	Alg.	O.S	300%					times of win				
			F-min (%)	F-maj (%)	o-acc. (%)	G-mean (%)	TPrate (%)	F-min (%)	F-maj (%)	O-acc. (%)	G-mean (%)	TP rate (%)
Breast-w	C4.5	NDO	93.04	96.13	95.02	95.36	96.47	2	2	2	2	3
		SMO	93.52	96.46	95.42	95.52	95.85	1	1	1	1	0
		RO	92.40	95.83	94.61	94.71	95.02	0	0	0	0	0
	NB	NDO	94.38	96.89	95.99	96.35	97.51	-	-	-	-	-
		SMO	94.38	96.89	95.99	96.35	97.51	-	-	-	-	-
		RO	94.38	96.89	95.99	96.35	97.51	-	-	-	-	-
	3-NN	NDO	96.39	98.00	97.42	97.96	99.75	-	-	-	2	2
		SMO	96.39	98.00	97.42	97.92	99.59	2+-	2+-	2+-	1	1
		RO	95.44	97.48	96.76	97.16	98.48	0	0	0	0	0
Diabetes	C4.5	NDO	66.86	77.76	73.39	74.16	76.94	3	3	3	3	1
		SMO	66.34	73.72	70.49	72.80	83.33	0	0	0	0	2
		RO	60.84	76.26	70.44	69.27	65.80	0	0	0	0	0

Table 2. (2). (Continued)

	NB	NDO	66.44	74.95	71.32	73.24	81.38	2	1	1	1	2	
		SMO	66.33	74.48	70.96	73.03	81.97	1	1	1	1	1	
		RO	66.39	76.14	72.09	73.50	78.98	0	1	1	1	0	
	3-NN	NDO	73.53	76.92	75.34	78.70	98.13	2	2	2	2	3	
		SMO	67.82	73.55	70.96	73.73	87.69	1	1	1	1	0	
		RO	61.43	68.79	65.49	67.81	78.73	0	0	0	0	0	
	Glass	C4.5	NDO	94.74	91.53	99.59	99.22	99.59	-	-	-	-	-
			SMO	94.74	94.74	99.76	99.53	99.76	1+-	1+-	1+-	1+-	-
			RO	94.74	94.74	99.76	99.53	99.76	-	-	-	-	-
NB		NDO	68.44	64.98	97.58	95.47	97.61	0	0	0	2	3	
		SMO	69.44	71.64	98.44	97.04	93.05	3	3	3	1	0	
		RO	64.86	64.00	97.77	95.79	92.42	0	0	0	0	0	
3-NN		NDO	73.30	73.77	98.41	97.01	98.43	1	2	2	3	3	
		SMO	73.85	67.57	98.02	96.26	94.49	2	1	1	0	0	
		RO	71.64	65.75	97.94	96.11	92.58	0	0	0	0	0	
Ionosphere	C4.5	NDO	86.22	92.80	90.54	88.54	82.46	2	2	2	0	0	
		SMO	85.50	90.87	88.79	89.48	92.06	0	0	0	2	3	
		RO	83.22	90.82	88.13	86.65	82.01	1	1	1	1	0	
	NB	NDO	82.91	90.07	87.44	86.86	84.92	3	3	3	2	0	
		SMO	82.71	89.45	86.89	86.98	87.30	0	0	-	1	1	
		RO	76.62	83.74	80.82	82.13	87.57	0	0	-	0	2	
	3-NN	NDO	95.42	97.31	96.61	97.01	98.49	2	2	2	2	1	
		SMO	95.04	97.04	96.30	96.86	98.94	1	1	1	1	2	
		RO	84.99	92.43	89.93	87.22	79.37	0	0	0	0	0	
Satimage	C4.5	NDO	67.20	95.70	92.40	86.67	80.16	2	2	2	3	3	
		SMO	65.13	95.59	92.17	84.08	75.20	1	1	1	0	0	
		RO	57.04	95.30	91.53	74.17	57.81	0	0	0	0	0	
	NB	NDO	47.15	88.34	80.89	83.84	87.68	0	0	0	1	3	
		SMO	48.82	89.31	82.32	84.27	86.77	3	3	3	2	0	
		RO	47.49	88.48	81.10	84.05	87.89	0	0	0	0	2	

Table 2. (2). (Continued)

Segment-challenge	3-NN	NDO	74.18	96.11	93.23	96.18	100.00	2	2	2	2	2		
		SMO	69.51	95.06	91.50	95.02	99.63	1	1	1	1	1		
		RO	64.87	94.80	90.94	88.73	86.08	0	0	0	0	0		
	Segment-challenge	C4.5	NDO	97.75	99.64	99.38	99.07	98.63	2	2	2	2	2	
			SMO	97.18	99.55	99.22	98.66	97.89	1	1	1	0	0	
			RO	97.24	99.56	99.24	98.53	97.56	0	0	0	1	1	
		Segment-challenge	NB	NDO	67.19	92.47	87.75	89.41	91.76	3	3	3	3	2
				SMO	61.15	89.03	82.89	89.01	98.54	0	0	0	0	0
				RO	60.81	88.78	82.56	88.98	99.02	0	0	0	0	1
Segment-challenge			3-NN	NDO	97.59	99.61	99.33	99.59	99.95	3	3	3	3	3
				SMO	97.45	99.59	99.29	99.38	99.51	0	0	0	0	0
				RO	96.23	99.38	98.93	99.24	99.67	0	0	0	0	0
	Sonar		C4.5	NDO	84.17	81.14	82.79	82.51	98.14	3	3	3	3	3
				SMO	82.12	79.93	81.09	81.07	93.13	0	0	0	0	0
				RO	69.86	75.15	72.76	72.28	67.70	0	0	0	0	0
		Sonar	NB	NDO	73.89	65.02	70.10	68.73	90.72	3	1	3	2	3
				SMO	70.43	65.39	68.11	67.81	81.44	0	1	0	0	0
				RO	70.33	63.20	67.15	66.43	83.51	0	1	0	1	0
Sonar			3-NN	NDO	89.27	88.49	88.89	89.03	99.07	2	2	2	2	3
				SMO	88.65	87.93	88.30	88.45	97.94	1	1	1	1	0
				RO	87.94	87.70	87.82	88.01	95.19	0	0	0	0	0
	Vehicle		C4.5	NDO	86.70	95.57	93.36	92.90	92.06	3	3	3	3	3
				SMO	85.69	95.35	92.99	91.67	89.28	0	0	0	0	0
				RO	85.55	95.33	92.95	91.46	88.78	0	0	0	0	0
		Vehicle	NB	NDO	56.46	71.73	65.72	73.30	94.47	2	1	1	1	2
				SMO	56.45	71.90	65.84	73.34	94.14	1	2	2	2	1
				RO	56.38	71.71	65.68	73.24	94.30	0	0	0	0	0
Vehicle			3-NN	NDO	88.12	95.67	93.66	95.76	100.00	2	2	2	3	3
				SMO	87.29	95.45	93.30	94.82	97.82	1	1	1	0	0
				RO	84.87	94.49	91.92	93.40	96.31	0	0	0	0	0

Note: “n+” denotes winning n times, and drawing with the other two approaches the left 3-n times; ‘-’ indicates drawing with the other one or two over-sampling approaches

The results described in table 2 reveal that the performance of each over-sampling strategy varies when implementing different baseline algorithms on a same data set, and the over-sampling rate influences the performance of an over-sampling approach too. We select dataset Diabetes as an example, if the over-sampling rate is set to 100%, NDO-Sampling outperforms SMOTE and RO-Sampling in terms of the listed five evaluation metrics when using C4.5 as the baseline classifier, but SMOTE performs the best in terms of the first four metrics when NB is executed, and NDO-Sampling win in term of TP rate. When the baseline algorithm is fixed to 3-NN, we find that SMOTE outperforms NDO-Sampling on the first four metrics when the over-sampling rate is set to 100%, but it lost when the over-sampling rate is set to 200%. The results reveal that the metric values are influenced by both the over-sampling rate and the classification algorithm. The above phenomenon occurs on almost all the datasets listed in table two.

The results shown in table 2 indicate that NDO-Sampling performs well against imbalanced data sets with continuous attributes. In many cases, our over-sampling strategy yields results in terms of the *F-measure* for minority, the *G-mean* and *TP rate* comparable or slightly higher than that produced by the other two approaches. It also achieves good results when the *F-measure* for majority class and the *overall accuracy* are considered. For the highest dimensional data set Sonar, the five metrics surpass that of SMOTE and random over-sampling all the time when C4.5 is conducted, the minority class *F-measure*, the *overall accuracy* and *TP rate* surpass that of SMOTE and random over-sampling all the time when NB is conducted. Our approach lost in terms of the five metrics only when the over-sampling rate is at 100% when implementing 3-NN. For the highly imbalanced data set Satimage, NDO-Sampling yields good results in terms of the five metric when C4.5 is implemented except that, it lost in terms of both of the *F-measures* and the *overall accuracy* when the over-sampling rate is set to 100%. SMOTE performs well when NB is conducted. When 3-NN is conducted, NDO-Sampling wins two times in terms of the five metrics except when the over-sampling rate is at 100%.

In order to facilitate the analysis of the results on the 8 data sets, we summarized the numbers of wins in table 2 for the three over-sampling approaches and show the results in table 3.

Table 3. Number of wins for 8 datasets with continuous attribute when implementing c4.5, NB and 3-NN

Alg.	O.S	F-min	F-maj	O-acc.	G-mean	TP rate
C4.5	NDO	20	17	16	16	15
	SMO	3	3	3	5	5
	RO	1	1	1	2	1
NB	NDO	11	8	9	9	11
	SMO	8	10	9	7	3
	RO	0	2	1	2	5
3-NN	NDO	13	14	14	18	18
	SMO	10	9	9	6	5
	RO	0	0	0	0	0

The results described in table 3 show that NDO-Sampling outperforms both SMOTE and RO-Sampling in terms of the *F-measures* of both the minority and majority class, the *overall accuracy*, *G-mean* and *TP rate* when C4.5 and KNN ($k=3$) are conducted. NDO-Sampling outperforms both SMOTE and RO-Sampling in terms of the four evaluation metrics (*F-measure for minority*, *Overall accuracy*, *G-mean* and *TP rate*) when NB is conducted, it only loses to SMOTE in *F-measure* for majority.

RO-Sampling performs worst among the three over-sampling strategies regardless of what classification algorithms are implemented. It loses all the time when 3-NN is conducted, because when RO-Sampling is employed to generate synthetic samples, it just duplicates the original samples randomly, and no real new samples are generated. When implementing NDO-Sampling and SMOTE to generate synthetic samples for the minority class, newly generated samples may become the nearest neighbors of a newly coming pattern, thus resulting in the increasing of the classification accuracy of KNN.

From the experimental results described in tables from table two and table three, we can conclude that NDO-Sampling has good potential in handling imbalanced data regardless of what learning algorithms have been used. NDO-Sampling also has good potential in handling imbalanced datasets with discrete attributes. Duo to the limit to the range of pages, the related work was not shown in this paper.

In order to evaluate the performance of NDO-Sampling on multiple-class classification problems, we conduct experiments on the original data set Satimage. Different over-sampling approaches are implemented on the class with the least number of samples. As the metrics such as *G-mean*, *F-measure* and *TP rate* have no meanings for multi-class classification problems, we just obtain the overall classification accuracies when different baseline algorithms are executed at different over-sampling rates. The averaged results of three experiments are shown in figure one.

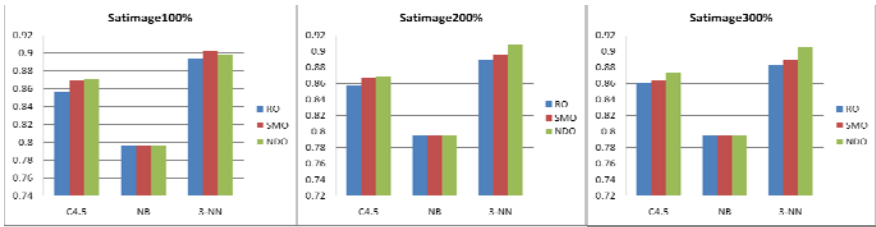


Fig. 1. Overall accuracies of multiple class classification on Satimage for different over-sampling strategies

For any given classification algorithm, obvious differences are not found in the overall accuracies at different over-sampling rates from figure two, because over-sampling is implemented on the minority class and the overall classification accuracies are mainly dominated by the other classes. Different over-sampling strategies obtain different classification accuracies, and NDO-Sampling performs the best most of the time.

5 Conclusions

Compared with SMOTE, NDO-Sampling reduces the computational complexity greatly, as it does not need to perform the nearest neighbor algorithm, a time-consuming approach, to obtain the k nearest neighbors before creating synthetic samples. SMOTE generates synthetic samples on the line between two neighbors, it can be considered as a linear interpolation approach. The newly generated samples does not hold the conclusions of the theorem one and two simultaneously, a necessary condition for the synthetic data and the original data to obey the same probability distribution. Random over-sampling re-samples the original minority class data with replacement, it is at the risk of over-fitting, but NDO-Sampling can avoid increasing the likelihood of occurring over-fitting. The key idea of NDO-Sampling is that it generates synthetic data that share an approximately similar probability distribution with that of the original minority data. In this paper, we just focus on the approach of over-sampling the minority class and ignore what a specific learning algorithm is conducted on the data sets. In our future work, considerations will be taken on algorithm-related normal distribution over-sampling approaches, such as just performing NDO-Sampling on the border samples for some specific algorithms.

Acknowledgments. This research is partially supported by the National Natural Science Foundation of China (No. 61170145), the Science and Technology Projects of Shandong Province, China (ZR2010FM021, and 2010G0020115) and Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology.

References

1. Barandela, R., Sanchez, J.S., Garcia, V., Rangel, E.: Strategies for learning in class imbalance problems. *Pattern Recognition* 36, 849–851 (2003)
2. Zhou, Z.-H., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18, 63–77 (2006)
3. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD Explorations Newsletter* 6(1), 20–29 (2004)
4. Sun, A., Lim, E.-P., Liu, Y.: On Strategies for Imbalanced Text Classification Using SVM: A Comparative Study. *Decision Support Systems* 48(1), 191–201 (2009)
5. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 301–312 (2006)
6. Yen, S.-J., Lee, Y.-S.: Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications* 36, 5718–5727 (2009)
7. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)
8. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC* 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)

9. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. SIGKDD Explorations Newsletter 6, 30–39 (2004)
10. Estabrooks, A., Jo, T., Japkowicz, N.: A Multiple Resampling Method for Learning from Imbalanced Data Sets. Computational Intelligence 20(1), 18–36 (2004)
11. Peng, Y., Yao, J.: AdaOUBOost: Adaptive Over-sampling and Under-sampling to Boost the Concept Learning in Large Scale Imbalanced Data Sets. In: MIR 2010, Philadelphia, Pennsylvania, USA, pp. 111–118 (March 2010)
12. Drummond, C., Holte, R.C.: C4.5, Class Imbalance and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. In: Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets (2003)
13. Zheng, Z., Wu, X., Srihari, R.: Feature Selection for Text Categorization on Imbalanced Data. SIGKDD Explorations Newsletter 6(1), 80–89 (2004)
14. Chen, M.-C., Chen, L.-S., Hsu, C.-C., Zeng, W.-R.: An information granulation based data mining approach for classifying imbalanced data. Information Sciences 178, 3214–3227 (2008)
15. Wu, G., Chang, E.Y.: KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on Knowledge and Data Engineering 17(6), 786–795 (2005)
16. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 445–453. Morgan Kaufmann, San Francisco (1998)
17. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179–186. Morgan Kaufmann, San Francisco (1997)

A Novel Genetic Algorithm for Overlapping Community Detection

Yanan Cai, Chuan Shi, Yuxiao Dong, Qing Ke, and Bin Wu

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia
Beijing University of Posts and Telecommunications, Beijing, China
{caiyanan, shichuan, dongyuxiao, keqing, wubin}@bupt.edu.cn

Abstract. There is a surge of community detection on complex network analysis in recent years, since communities often play special roles in the network systems. However, many community structures are overlapping in real world. For example, a professor collaborates with researchers in different fields. In this paper, we propose a novel algorithm to discover overlapping communities. Different from conventional algorithms based on node clustering, our algorithm is based on edge clustering. Since edges usually represent unique relations among nodes, edge clustering will discover groups of edges that have the same characteristics. Thus nodes naturally belong to multiple communities. The proposed algorithm apply a novel genetic algorithm to cluster on edges. A scalable encoding schema is designed and the number of communities can be automatically determined. Experiments on both artificial networks and real networks validate the effectiveness and efficiency of the algorithm.

Keywords: community detection, overlapping community, genetic algorithm, link community.

1 Introduction

Nowadays, community detection, as an effective way to reveal the relationship between structure and function of networks, has drawn lots of attention and been well developed. To do so, networks are modeled as graphs, where nodes represent objects and edges represent the interactions among them. Community detection divides a network into groups of nodes, where nodes are densely connected inside, while sparsely connected outside. However, in real world, 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. All of these objects represent the interaction between communities and then play an important role in the stability of the network. In community detection, these objects should be divided into multiple groups, which is known as overlapping. Overlapping community detection still remains a challenge in community detection.

Until now, lots of overlapping communities have been proposed, which can be roughly divided into two classes, node-based and link-based overlapping community detection algorithms. The node-based overlapping community detection

algorithms [1,2,3,4,5,6,7,9], classify nodes of the network directly. The link-based algorithms cluster the edges of network, and map the final link communities to node communities by simply gather nodes incident to all edges within each link communities [8]. All of these algorithms contribute to overlapping community detection, however, they still have disadvantages. For example, the coverage of CPM [2] largely depends on the feature of the network, etc.

In this paper, we propose a genetic algorithm to detect overlapping community with link clustering, which is named Genetic algorithm for overlapping Community Detection (GaoCD). The algorithm first finds the link communities by optimizing objective function partition density D [8], and then map the link communities to node communities based on a novel genotype representation method. The number of the communities found by GaoCD can be automatically determined, without any prior information. Experiments on both artificial networks and real networks are designed to validate the algorithm. Experiment on artificial networks shows that GaoCD work well on networks with typical overlapping structure. Experiments on real networks compare GaoCD with ABL [8] and GA-NET+ [9]. It is shown that GaoCD always achieves higher partition density D and finds denser communities.

The paper is organized as follows. In the next section, we introduce the related works. Section 3 describes the new genetic algorithm we propose, including framework, objective function, genetic representation, and operators. The experimental results are illustrated in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Many algorithms have been developed to detect overlapping communities in complex networks, such as CPM [2], CONGA [5], GA-Net+ [9], etc. Among them, CPM is the most famous and widely used. However, CPM has a strict community definition and is not flexible enough for real network. When the network is too dense, CPM finds giant clique communities, however, when the network is too sparse, it finds no cliques at all. And thus, the coverage of CPM largely depends on the feature of the network, providing no global prospective for the whole network.

GA-Net+ [9], proposed by Pizzuti, first adopts genetic algorithm to detect overlapping communities. It proposes a method to transfer node graph to line graph, in which nodes present edges of the node graph, while edges present adjacent relationships of edges of node graph. The line graph is then used as the input of the genetic algorithm, and in each generation, the line graph is transferred to node graph to evaluate the fitness. After selection, the graph is transferred again for the next iteration of GA. The transfer between line graph and node graph costs much computation and decreases the effectiveness. GaoCD is also a genetic algorithm, but it clusters edges of the network, with different genotype, objective function and operators. Instead of community score [9], GaoCD adopts partition density D to evaluate the quality of the partition.

What's more, the partition found by GaoCD coverages the whole network and provides a global view of the structure of the network.

Recently, link based methods are proposed to detect overlapping communities. Based on the thought that each edge plays an unique role in the network, Ahn, Bagrow and Lehmann [8] first propose a link-based algorithm, clustering the edges of the network. They define the similarity of edges and an evaluation function for link community, partition density D . The algorithm first calculates the similarity of all edges of the network and assign each edge to its own community. At each step, the method chooses pairs of edges with the largest similarity and merges their respective communities until all edges belong to a single cluster. Then, the history of the clustering process is stored in a dendrogram, and the partition with the largest partition density D is chosen as the final result. As shown is section 4, this algorithm tends to find small communities and can not provide global view of the structure of the network.

There are other algorithms for overlapping community detection, such that the SCP of Kumpula [10], Lancichinetti's algorithm [7], etc. All of them need prior information, or have coverage problem, or suffer of efficiency.

3 The New Genetic Algorithm for Community Detection

In this section, we discuss our algorithm in detail, including the framework of the algorithm, objective function, the crucial genetic representation and operators.

3.1 Framework of the Algorithm

Genetic algorithm, derived from evolutionary biology, is a searching technique to find exact or approximate solutions for optimization problems. The GA algorithms are implemented as computer simulation, in which a population of abstract representations of candidate solutions to an optimization problem evolves towards approximate solutions, based on the production and selection schema. The framework of GaoCD is described in Algorithm 1.

To effectively apply genetic algorithm to solve overlapping community detection problem, we design a new kind of genetic representation, encoding the edges of the network and a specific decoding schema taking the edge feature into consideration. The genetic representation and operation designed effectively reduce the search space and thus improve the searching effectiveness.

3.2 Objective Function

GaoCD is a link-based algorithm, which finds link communities. For this reason, the novel genetic algorithm chooses link community evaluation function, partition density D , as the objective function. Partition density D is raised by Ahn in [8], evaluating the link density within the community, as described in Equation (1).

$$D(c) = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c - 1)}{2} - (n_c - 1)} \quad (1)$$

Algorithm 1. Main framework of GaoCD

```

1: procedure GAOCD(size, gens, pc, pm)
2:   // size is the size of the population.
3:   // gens is the running generation.
4:   // pc and pm are the ratio of crossover and mutation, respectively, with  $p_c \in [0, 1]$ ,  $p_m \in [0, 1]$  and  $p_c + p_m = 1$ .
5:    $P_t = \Phi$ 
6:   for each i in 1 to size do
7:      $g_i = \text{generate\_individual}()$ 
8:      $\text{evaluate}(g_i)$ ;  $P_t = P_t \cup \{g_i\}$ 
9:   end for
10:  for each gen  $\leftarrow t$  in 1 to gens do
11:     $i = 0$ ;  $P_{t+1} = \Phi$ 
12:    while  $i < \text{size}$  do
13:      randomly select two individuals from  $P_t$ ,  $g_j$  and  $g_k$ ,  $j, k \in [1, \text{size}]$ 
14:      generate random value  $r \in [0, 1]$ 
15:      if  $r < p_c$  then
16:         $g'_j, g'_k = \text{crossover}(g_j, g_k)$ 
17:      else  $g'_j = \text{mutate}(g_j)$ ;  $g'_k = \text{mutate}(g_k)$ 
18:      end if
19:       $i = i + 2$ 
20:       $\text{evaluate}(g'_j)$ ;  $P_{t+1} = P_{t+1} \cup \{g'_j\}$ 
21:       $\text{evaluate}(g'_k)$ ;  $P_{t+1} = P_{t+1} \cup \{g'_k\}$ 
22:    end while
23:     $\text{selection}(P_{t+1}, P_t \cup P_{t+1})$ 
24:  end for
25:  return  $P[1]$ 
26: end procedure

```

$\text{generate_individual}()$ //initialize an individual according to the genetic representation schema.
 $\text{evaluate}(g)$ //evaluate the fitness of g individual according to objective function partition density D .
 $\text{crossover}(g_j, g_k)$ //crossover genetic operator.
 $\text{mutate}(g_j)$ //mutation genetic operator.
 $\text{selection}(P_{t+1}, P_t \cup P_{t+1})$ //The selection step of the genetic algorithm. First select *size* individuals with maximum fitness from $P_t \cup P_{t+1}$, and then fill in P_{t+1} one by one in decreasing order according to fitness value.

Define $P = \{P_1, \dots, P_C\}$ 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 partition density D is the average of D_c over all communities, weighted by the fraction of links present in each:

$$D = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \quad (2)$$

As we can see that partition density D only considers the link density within the community, different from the common community definition that a community should be densely intra-connected and sparsely connected with the rest communities. For overlapping communities, this definition make no sense, as Fig. 1 shows. In this figure, there are three obvious communities, all of which are cliques. Because all of the communities are overlapping, with node 0 as common node, each community is densely intra-connected, while not sparsely connected with other communities.

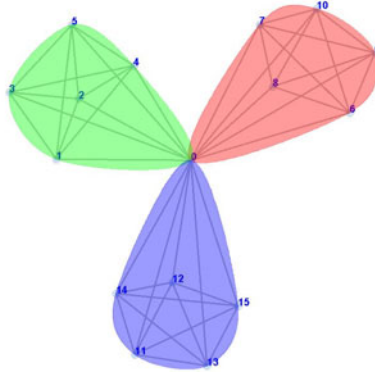


Fig. 1. A classical network containing overlapping communities

3.3 Genetic Representation

In this section, we describe the genetic representation in detail, including the encoding and decoding phase.

Encoding Phase. For those genetic algorithms for community detection which encoding the nodes of the network, we refer to them as node-based, such as GACD [11], GA-Net+ [9]. Different from the node-based genotype, GaoCD encodes the links of the network. In this link-based representation, an individual g of the population consists of m genes $\{g_0, g_1, \dots, g_i, \dots, g_{m-1}\}$, where $i \in \{0, \dots, m-1\}$ 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 . According to graph theory, two edges are adjacent if they share one node in undirected graph. For example, in Fig. 2 (a), edge 0 has four adjacent edges, 1, 5, 2, 6, and one possible value for g_0 is 1, as shown in Fig. 2 (b).

Decoding Phase. The decoding phase transfers an 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

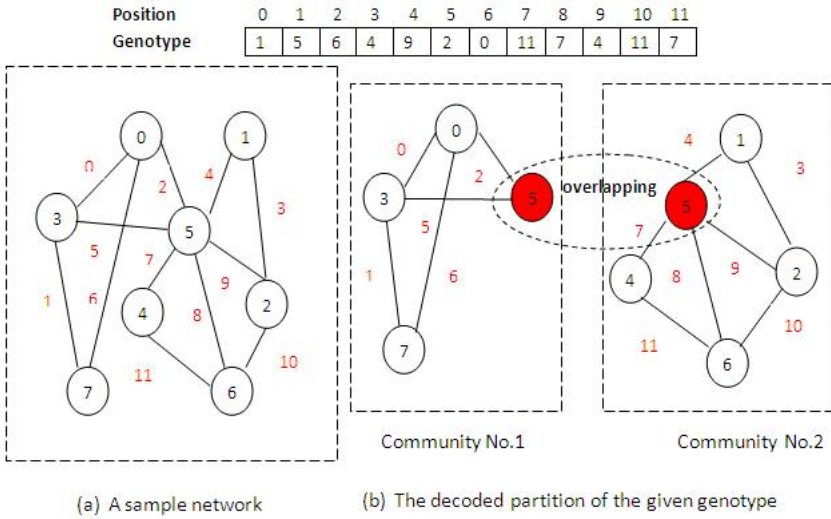


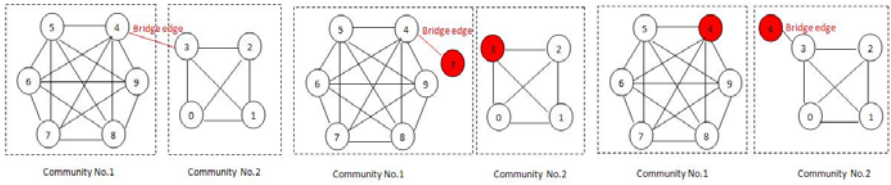
Fig. 2. Illustration of the genetic representation;(a) A simple network for encoding;(b) A possible genotype for the network in (a) and the corresponding decoded partition

found, and all edges within the component constitute a link community. According to link-based algorithm, by Ahn [8], overlapping communities are contained simply by gathering the nodes incident to all edges for each link community. However, it is not suitable for our algorithm, which restrain that each link community contains more than one link for the purpose of full coverage. We raise a fine tuning schema to deal with a special kind of edges, which is called **Bridge Edge**.

Bridge Edge is defined as the edge connecting two communities, as Fig. 3(a) edge (3,4) shows. It is obvious that Fig. 3 contains two absolute communities, both of which are cliques. Because the bridge edge must belong to one unique link community, then the bridge edge could to classified to any of the two cliques, as shown in Fig. 3(b) and (c). By simply gathering the nodes incident to edges of link community to form node community, we obtain overlapping communities shown in Fig. 3(b) and (c) respectively, which obviously opposite to our purpose. To avoid this problem, we raise fine tuning method.

Fine tuning adjust the node membership of node community obtained by simple mapping schema. It is designed for nodes which have multi membership. The method first finds the list of nodes which have multi membership, and then for each node i in the list, it tests that whether node i contributes to the communities $c_{i1}, c_{i2}, \dots, c_{in}$ by adding to them, where c_{ij} is community containing node i . Here, we adopt average degree of the community to evaluate the contribution. The definition of average degree is as follows:

$$AD(c) = 2 * \frac{|E(c)|}{|c|} \tag{3}$$



(a) reasonable partition (b) one possible unreasonable partition (c) the other possible unreasonable partition

Fig. 3. Illustration of bridge edge problem(edge of red color represents bridge edge);(a) A sample network containing bridge edge (3,4); (b) A possible partition found by the simple mapping schema from link community to node community, with node 3 classified to both communities. (c) The other possible partition found by the simple mapping schema from link community to node community, with node 4 classified to both communities.

where c is a community, $E(c)$ is the number of edges in the community, and $|c|$ is the number of nodes of the community.If adding to the community makes $|AD(c)|$ increase, we suppose that the node contributes to the community. If node i contributes to community c_{ij} , then the community stays the same(note that the community contains the node originally), otherwise, the node is removed from the community. If average degrees of all the n communities the node belongs to decrease with the node in, the least decreased community stay same, while others remove the node from the community. In Fig. 3 (b), node 3 is overlapping and contained in community No.1 and community No.2. Because node 3 decreases the $|AD(c)|$ of community No.1, while increases the $|AD(c)|$ of community No.2, then node 3 should be removed from community No.1, which is the most reasonable partition. In Fig. 3 (c), node 4 decreases the $|AD(c)|$ of community No.2, while increases the $|AD(c)|$ of community No.1, and then node 4 would be removed from community No.2.

3.4 Operators

In GaoCD, we assume that the network is a simple undirected connected graph. According to the genetic representation, we further raise the corresponding operators. Both the crossover and mutation ensure that the edge corresponding to gene i is incident to edge i . Suppose that we have two genotypes $g1$ and $g2$, and in the crossover phase, a random value i is generated. If the value of the i th gene of them are j and k respectively, through crossover, the value of i th gene exchanged, which makes the i th gene of genotype $g1$ is k , and the i th gene of genotype $g2$ is j . Because i th edge is incident to both edge j and edge k according to the generation of $g1$ and $g2$, the crossover has no effect on the principle that the edge of gene i should be incident to the i th edge of the network. In the mutation phase, each random value is generated for each parent, please notice that the values may not equal. If the value generated is i for genotype $g1$, and

the i th edge is incident to edge i_1, \dots, i_n , then the i th value of $g1$ is replaced by $i_k, k \in 1, \dots, n$. Fig. 4 shows the operation of network in Fig. 2 (a).

3.5 Discussion

GA is a search technique for optimization problem. When applied to specific problem, it is essential to design an appropriate genotype, including the encoding schema and decoding schema. The encoding schema we design ensures the algorithm covers the whole network, leaving no isolated nodes. The decoding schema of GaoCD probably deal with the bridge edges of the network, which improves the accuracy. What’s more, the encoding schema reduces the search space from $O(E^E)$ to $O(d^E)$, where d is the number of incident edges of each edge, E is the length of genotype, $d \ll E$. Reducing the search space makes GaoCD search the more accurate solution with less consuming time.

4 Experiments

In order to test the effectiveness of GaoCD, we design experiments on artificial networks and real networks respectively. The experiment on artificial networks evaluate the ability of GaoCD to discover the overlapping nodes on different kinds of networks. The experiment on real networks compares GaoCD with ABL [8] and GA-NET+ [9]. All of the experiments are carried out on a 2.66GHz and 2G RAM Pentium IV computer.

4.1 Experiments on Artificial Networks

We first use artificial networks to test the performance of our algorithm. Fig. 5 includes several artificial networks, each of which represents a type of network structure. Testing all these tiny networks provide a view of the complicated networks with similar structure. In Fig. 5, each color represents one community. We

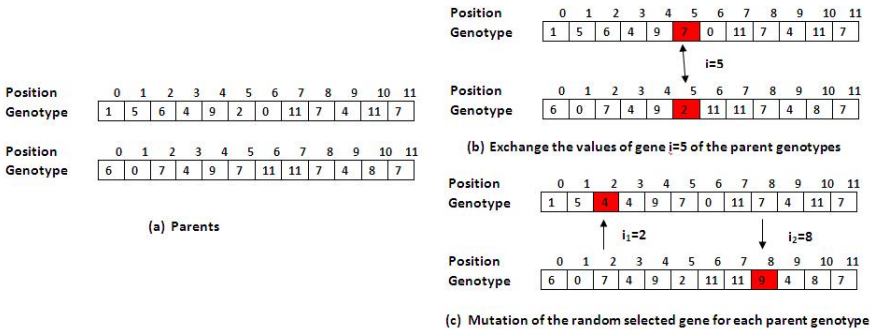


Fig. 4. Illustration of operators. (a) shows two genotypes of the population; (b) is one possible crossover of parent genotype of (a); (c) show a possible mutation for each parent genotype;

can see that network a and network c contain bridge edges which should not be divided into either of the communities in theory, GaoCD successfully distinguish the bridge edges and deal with them properly. Network b is a hierarchical core network, a node could be the core of a community, and at the same time, it belongs to another core community with another node as the core. In network d , GaoCD correctly divided the sharing nodes of the two cliques to both them. Overall, GaoCD correctly finds the overlapping communities for all kinds of networks, and ensures all nodes of the networks are covered in the partition, leaving no absolute nodes.

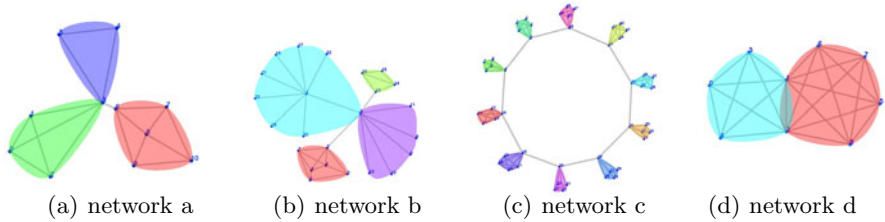


Fig. 5. Four typical kinds of networks

4.2 Experiments on Real Networks

In this section, we first validate GaoCD on real networks with partition density D as the evaluation function, compared with ABL and GA-NET+. And then, we investigate the partition found by GaoCD by analyzing the community sizes of the partition. At last, an intuitive view is given for the partition found by GaoCD.

As stated by Ahn [8], when overlap is pervasive, each community has many more external than internal connections, the common definition is not suitable. Here we adopt partition density D as the evaluation function, which only considers the link density inside the community. Here, we validate GaoCD on several common data sets, as described in Table 1.

Table 1. Real networks

	karate(N1)	polbooks(N2)	dolphins(N3)	football(N4)	lesmis(N5)
Nodes	34	105	62	115	77
Edges	78	441	159	613	254

As shown in Fig. 6, for all real networks, GaoCD has the highest partition density D , which means that the partition found by GaoCD is denser than ABL and GA-Net+ does. To further investigate the quality of the partition found by GaoCD, we analyze the community size distribution of the communities for all real networks. For communities with size two contains single link, representing bridge edges or isolated nodes, which do not contribute to partition density D . We classify the communities into three classes, the first contains communities

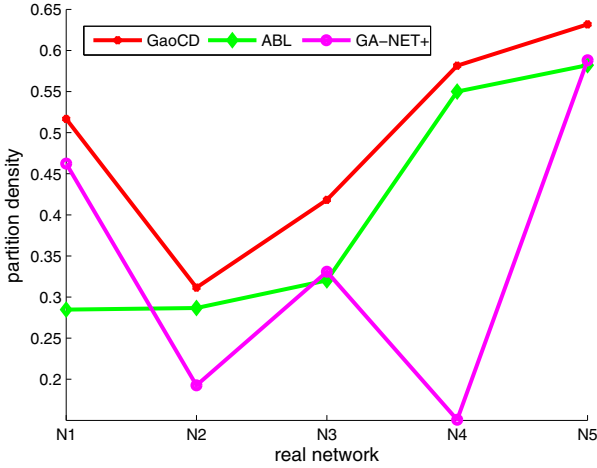


Fig. 6. Comparison of GaoCD, ABL, GA-Net+, relative to partition density D for real networks

with sizes varying from three to five, and second from six to ten, communities larger than ten belong to the third class. Fig. 7 shows the ratio of all three classes over all communities, respectively. It is easy to see that ABL tends to find small communities, with size from three to five. For almost all networks, GaoCD has larger values for ratio in middle and large classes. Overall, ABL tends to find tiny communities, and can not reflect the whole structure of the network. While GaoCD, on the contrast, finds denser communities in all sizes, capturing macrostructure as well as the microstructure of the network.

Fig. 8 show the partitions found by GaoCD. Fig. 8 (a) is the co-purchasing network of books about US politics by the online bookseller Amazon.com. Nodes represent books, and the edges between nodes represent frequent co-purchasing of books by the same buyers. It is obvious that the network constitutes of two large communities, with each of them surrounded by small ones. Node 6, node

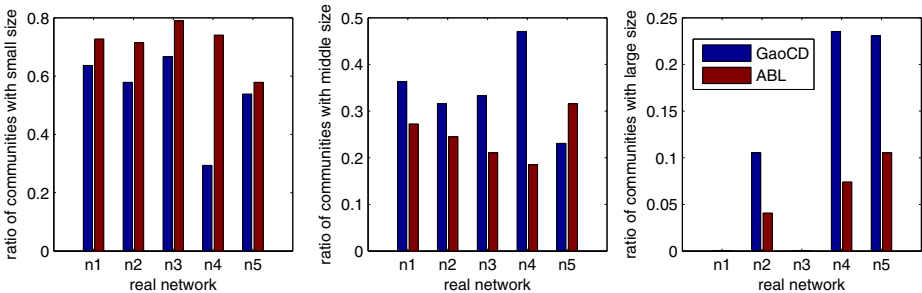
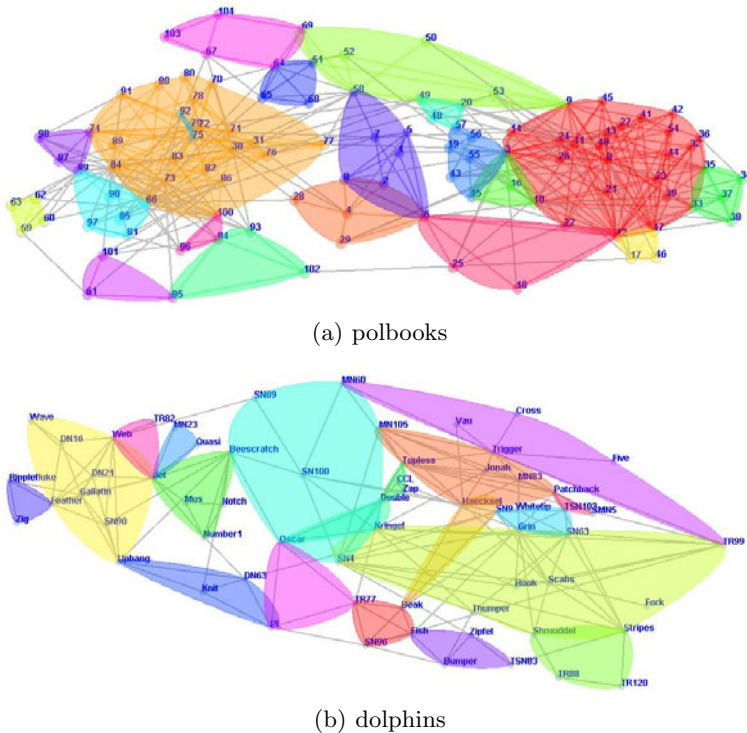


Fig. 7. Comparison of GaoCD, ABL, GA-Net+, relative to partition density D for real networks



(a) polbooks

(b) dolphins

Fig. 8. Partitions found by GaoCD.(a) is for network polbooks, and (b) for network dolphins.

58 and node 3 are nodes with obvious overlapping membership. Fig. 8 (b) is a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand. The network fell into two parts because of the living of SN100. GaoCD successfully distinguish the special role of SN100, finding SN100 as the core of a community, connecting nodes from other different communities. Removing SN100, the core of the community, makes other nodes of the community unconnected, which then splits the networks.

5 Conclusion

In this paper, we propose a genetic algorithm for overlapping community detection, optimizing partition density D . Different from those node-based overlapping community detection algorithms, GaoCD utilizes the property of the unique role of links and applies a novel genetic algorithm to cluster on links. The genetic representation and the corresponding operators significantly reduce the search space and make the number of the communities determined automatically. Moreover, GaoCD covers all nodes of the networks, no matter the

network is dense or sparse. To validate our algorithm, experiments on artificial networks and real networks are carried out, respectively. Both of them show that GaoCD finds overlapping structure successfully. Compared with ABL and GA-Net+, GaoCD finds denser communities, which reflects the macrostructure as well as the microstructure of the network.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No.60905025, 90924029, 61074128).

References

1. Pereira, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. *Proteins: Structure, Functions, and Bioinformatics* 54, 49–57 (2004)
2. 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)
3. Baumes, J., Goldberg, M., Magdon-Ismael, M.: Efficient Identification of Overlapping Communities. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) *ISI 2005. LNCS*, vol. 3495, pp. 27–36. Springer, Heidelberg (2005)
4. Zhang, S.H., Wang, R.S., Zhang, X.S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A* 374, 483–490 (2007)
5. Gregory, S.: An Algorithm to Find Overlapping Communities Structure in Networks. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) *PKDD 2007. LNCS (LNAI)*, vol. 4702, pp. 91–102. Springer, Heidelberg (2007)
6. Gregory, S.: A fast algorithm to find overlapping communities in networks. In: *PKDD*, pp. 408–423 (2008)
7. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure of complex networks (2008), arXiv:0802.1281, physics.soc-ph
8. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* 466, 761–764 (2010)
9. Pizzuti, C.: Overlapping Community Detection in Complex Networks. *ACM* (2009)
10. Kumpula, J.M., et al.: Sequential algorithm for fast clique percolation. *Phys. Rev. E* 78, 026109 (2008)
11. Shi, C., Yan, Z.Y., Wang, Y., Cai, Y.N., Wu, B.: A Genetic Algorithm for Detecting Communities in Large-scale Complex Networks. *ACS* 13(1), 3–17 (2010)

A Probabilistic Topic Model with Social Tags for Query Reformulation in Informational Search

Yuqing Mao^{1,2}, Haifeng Shen², and Chengzheng Sun¹

¹ School of Computer Engineering, Nanyang Technological University,
BLK N4, Nanyang Avenue, Singapore 639798, Singapore

² School of Computer Science, Engineering & Mathematics, Flinders University,
Adelaide, South Australia 5001, Australia

{mao.yuqing, haifeng.shen}@flinders.edu.au, czsun@ntu.edu.sg

Abstract. It is non-trivial to formulate a query that can precisely describe the goal of an informational search task. Query reformulation based on the query clustering approach addresses this issue by expanding a new query with related existing queries that were generated by other users. However, the query clustering approach is unable to cluster queries that are intrinsically related but neither contain common terms nor return common clicked Web page URLs. More importantly, it does not address the issue of ranking retrieved results according to their relevance to the search goal. In this paper, we present new query reformulation approach based on a novel probabilistic topic model to discovering the latent semantic relationships between the queries and the URLs. It can not only discover related queries that cannot be clustered by existing query clustering approaches but also rank retrieved results according to the similarities of probability distributions over the latent topics among the queries and the URLs. The results of our experiments have shown that this approach can significantly improve the performance of an informational search task in terms of search accuracy and search efficiency.

1 Introduction

An informational search task is intended to find information about a topic [12]. In contrary to a navigational search task, which is intended to find specific resources that the user has already had in mind (e.g., searching for books written by a specific author) and can be completed by generating a single or couple of easy-to-formulate queries, an informational search task usually requires to generate a set of queries and click to view a number of Web pages retrieved by each query. The main reason is that the user performing an informational search task is unlikely an expert in the topic domain and consequently it is non-trivial for them to formulate a query to precisely describe the task goal. As such, they have to iterate between generating queries and view the retrieved results until they have acquired the information that matches the task goal, i.e., results relevant to the topic, through the trial-and-error search process.

If a query does not precisely describe the goal of an informational search task, retrieving the results only matching the query terms and ranking them simply according

to the term frequency cannot achieve the task goal. As informational tasks are getting common in Web search, it is imperative that a query can retrieve Web pages that are relevant to the search goal, regardless of term match, and that are ranked according to their relevance to the search goal, regardless of term frequency.

One of the major technologies to addressing this issue is query reformulation [11]. Recent research has been focusing on the query clustering approach through analyzing the click-through information in query logs. The approach can be illustrated by a bipartite graph [3] in Fig 1, where vertices in the left set $\{q_1, q_2, q_3, q_4\}$ represent queries composed of terms, e.g., query q_1 is composed of a set of terms $\{t_{11}, t_{12}, \dots, t_{1m}\}$, vertices in the right set $\{u_1, u_2, u_3, u_4\}$ represent Web page URLs, and a solid edge between q_x to u_y represents u_y has been clicked by the user who issued q_x , while a dashed edge represents u_y is relevant to q_x according to the search goal but it is not retrievable by q_x through matching terms.

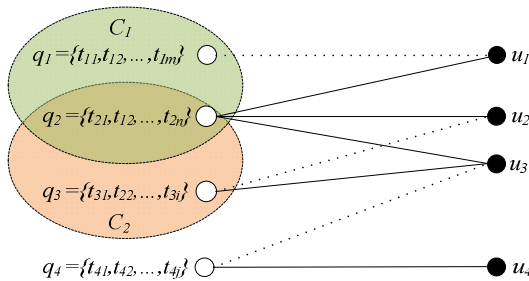


Fig. 1. Bipartite graph representation of click-through data

There are two main query clustering techniques. One is based on the similarity between queries measured by Levenshtein edit distance [15], as queries with common terms are likely to be related. In Fig 1, queries q_1 and q_2 are placed in the same cluster C_1 using this technique because both queries contain a common term t_{12} . The other is based on the relationships between queries and Web pages [3], as queries returning common clicked URLs or similar Web page contents are likely to be related. In Fig 1, queries q_2 and q_3 are placed in the same cluster C_2 using this technique because both queries return a common clicked URL u_3 .

However, this approach is unable to cluster queries that are intrinsically related but neither contain common terms nor return common clicked URLs. In Figure 1, suppose $q_4 =$ “how to speed up windows xp”, $q_1 =$ “anti virus update”, $q_2 =$ “security system virus”, and $q_3 =$ “download service pack 2”. Neither will q_4 be clustered into C_1 nor into C_2 because it does not contain any term in common with those used by the queries in C_1 or return any clicked URL in common with those clicked by the users who generated the queries in C_2 , although q_4 is intrinsically related to both q_1 and q_3 because in reality many users who generated q_4 also generated q_1 or q_3 .

More importantly, the query clustering approach does not address the issue of ranking retrieved Web page URLs according to their relevance to the goal of an informational search task. URLs are usually ranked according to the “popularity” – the number of clicks in a cluster [2]. Although some ranking techniques are available,

they are independent of the query clustering approach and generally do not rank results according to the task goal.

In this paper, we present a new query reformulation approach that is different from query clustering. The cornerstone of the new approach is a novel topic model to discover the latent semantic relationships between the queries and the URLs retrieved by the queries. The topic model is based on the statistics of co-occurrence of query terms and URLs, and a variety of topics are derived to measure the similarity among the queries based on their probability distributions over these topics. The novelty of this approach is twofold. One is that it can discover related queries that cannot be clustered by the query clustering approach, because it measures the relevance of queries at the term level (i.e., the probability of terms being associated with the same topic), while the query clustering approach does that at the query level (i.e., comparing queries in their entireties in order to find out common terms or URLs). In our approach, the queries are related if they have similar probability distributions determined by the probability distribution of each term in the queries. The other novelty of the approach is that it can rank the retrieved Web page URLs according to their relevance to the goal of an informational search task based on the similarity of the probability distributions among the URLs and the queries over all latent topics. Further, we model the process of informational search with social tags (e.g., social bookmarks) to infer users' actual goals so that more relevant topics could be derived.

We have conducted a set of experiments to validate this approach and the results have shown that our approach can improve the performance of an informational search task in terms of search accuracy and search efficiency defined in Section 4.

The rest of the paper is organized as follows. The related work is discussed in Section 2. Section 3 presents the topic model and its application to informational search. The experiments and their results are discussed in Section 4. Finally, Section 5 concludes the paper with our major contributions and planned future research work.

2 Related Work

Query reformulation has been studied extensively. One of the earliest notions is Rocchio's classic query reformulating scheme [11], where query terms were re-weighted based on feedback relevance. Recent research efforts include local context analysis based on pseudo-relevant documents [17] (e.g., top-ranked documents), mining term association rules for automatic global query reformulation based on selected corpus [13] (e.g., TREC collections). These efforts share one thing in common: they do not exploit the real search activities performed by real users, e.g., the queries they generated and the URLs they clicked.

User interactions recorded in user logs have been used to improve the performance of query reformulation [6]. In particular, recent research has been focusing on clustering queries based on the click-through information in query logs in order to discover the relevance of the queries frequently submitted to a search engine. The objective is to expand a new query with related existing queries that were generated by other users. In Beeferman's work [3], a bipartite graph was first constructed from the

click-through data to represent the queries and the retrieved documents and a graph-based agglomerative iterative clustering method was then used. In Baeza-Yates' work [2], a weighted graph derived from the query-click bipartite graph was used to infer the semantic relations among the queries in a vector space. Wen's work [15] used a density-based method to estimate the similarity between queries by combining query content and click-through information. Although these research efforts have exploited the real search activities performed by real users, some intrinsically related queries still could not be clustered together because they analyze query relevance at the query level and queries in their entireties are generally sparse. In contrast, our approach analyzes query relevance at the term level and therefore it can discover more related queries because terms are less sparse than queries.

Moreover, they do not rank the Web page URLs retrieved by the queries in a cluster according to their relevance to the general search goal of that query cluster. Instead, queries in the same cluster are treated equally and each query's results are ranked individually. Optimizing or re-ranking search results from multiple queries by analyzing click-through data is a separate research endeavor independent of query clustering [16, 18], such as "learning to rank" with Neural Network or Support Vector Machine. For example, in Joachims's work [8], a ranking function could learn from the implicit feedback in a search engine's click-through data to provide personalized ranking of search results. In contrast, ranking of search results from multiple queries according to their relevance to the general goal of an informational search task is built into our query reformulation approach using the same probabilistic topic model.

Our topic model is extended from LDA (Latent Dirichlet Allocation) [5] - a probability-based language model that can be used to find the latent semantic relationships between the words in a document. LDA and its extensions and variants have emerged as a useful family of models with many interesting applications mainly for natural language processing. Recently, there are some attempts to use LDA for information retrieval. For example, Wei and Croft [14] incorporated LDA into a language modeling framework to improve the performance of ad-hoc retrieval.

Our model RTU-LDA (Relevance for Terms and URLs) is a significant extension to the original LDA. In particular, our model has a hierarchical structure for recording an informational search process: a document comprises a number of queries; each query is associated with a number of clicked URLs; and each URL is labeled by a number of social tags.

3 The Probabilistic Topic Model for Query Reformulation

In this section, we will systematically present our topic model for query reformulation in informational search tasks, including the topic model, estimation of parameters in the model, and retrieval and ranking of URLs according to their relevance to the queries based on the probability distributions over the topics derived from the model.

3.1 Topic Model with Social Tags

As a significant extension of LDA, our topic model RTU-LDA allows a document to be hierarchically structured and tagged with any number of labels. Each document records a user’s informational search activity: first choosing terms to formulate a series of related queries, then clicking some of the retrieved URLs to view the Web pages, and finally adding some tags to annotate some of the viewed Web pages.

The graphical representation of the RTU-LDA model is shown in Fig 2, where the notations are given in Table 1:

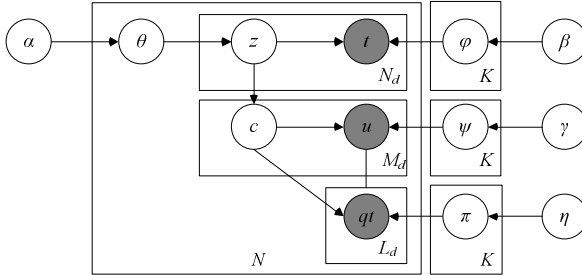


Fig. 2. RTU-LDA’s graphical representation

Table 1. Notations used in RTU-LDA

Symbol	Description	Symbol	Description
K	number of topics	z, c	Topic
qt	query term	L_d	number of query terms in document d
u	Web page URL	M_d	number of URLs in document d
t	Tag	N_d	number of tags in document d
θ	document’s topic distribution	η	query term hyperprior
π	topic’s query term distribution	γ	URL hyperprior
ψ	topic’s URL distribution	β	tag hyperprior
φ	topic’s tag distribution		
N	number of documents; each document is composed of the click-through information and the tags for the clicked URLs collected from an informational search process		

In RTU-LDA, query terms, URLs, and tags are all observed variables, while the hidden variable – the latent topic - can be discovered by the observation of tags in the training process. The generative process of a document can be abstracted from an informational search process: first selecting tags that can express the user’s search goal, then selecting URLs labeled with these tags, and finally selecting query terms used to retrieve these URLs. URLs not labeled with any tag and terms not associated with any clicked URL are selected with the probabilities specified by a distribution.

Formally, RTU-LDA assumes each document d has topic proportions θ_d that are sampled from a Dirichlet distribution. For each topic z , a collection of tags t_d are selected from a topic-specific multinomial distribution φ_z . Topic c is sampled from topics $z_d = \{z_{dn}\}_{n=1}^{N_d}$ with a multinomial distribution, in which $p(c = k) = \frac{N_{kd}}{N_d}$, where N_{kd} is

the number of tags that are assigned to topic k in the d th document. For each topic c , a collection of URLs u_d are sampled from a topic-specific multinomial distribution ψ_c , and a collection of query terms qt_d are sampled from a multinomial distribution π_c .

RTU-LDA assumes the following generative process for a collection of documents, each of which has the structure of $\{(t_d, u_d, qt_d)\}_{d=1}^D$.

1. Draw topic proportions $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For each tag t_{dn} , $n = 1, \dots, N_d$,
 - (a) Draw topic $z_{dn} \sim \text{Multinomial}(\theta_d)$; (b) Draw tag $t_{dn} \sim \text{Multinomial}(\varphi_{z_{dn}})$
3. For each URL u_{dm} , $m = 1, \dots, M_d$,
 - (a) Draw topic $c_{dm} \sim \text{Multinomial}(\{\frac{N_{kd}}{N_d}\}_{k=1}^K)$; (b) Draw URL $u_{dm} \sim \text{Multinomial}(\psi_{c_{dm}})$
4. For each query term qt_{dl} , $l = 1, \dots, L_d$,
 - (a) Draw topic $c_{dl} \sim \text{Multinomial}(\{\frac{N_{kl}}{N_d}\}_{k=1}^K)$;
 - (b) Draw query term $qt_{dl} \sim \text{Multinomial}(\mu_{c_{dl}})$

Based on the model, the likelihood of all query terms $QT = \{qt_d\}_{d=1}^D$ being associated with specific topics is the joint distribution of all variables (observed and hidden):

$$p(Z, T, C, U, QT | \alpha, \beta, \gamma, \eta) = p(Z | \alpha) p(T | Z, \beta) p(C | Z) p(U | C, \gamma) p(QT | C, U, \eta)$$

where $Z = \{z_d\}_{d=1}^D$, $T = \{t_d\}_{d=1}^D$, $C = \{c_d\}_{d=1}^D$, and $U = \{u_d\}_{d=1}^D$.

3.2 Estimation of Parameters

Inference of all latent topics Z from existing documents entails learning the model parameters θ , φ , ψ , π - the distributions over topics, tags, et al - respectively.

Although exact computation of these parameters is intractable, several approximation methods have been proposed to solve similar parameter estimation problems. We adopt Gibbs Sampling [7] - a special case of Markov-chain Monte Carlo methods that estimate a posterior distribution of a high-dimensional probability distribution - to solve the parameter estimation problem in the RTU-LDA model. The sampler draws from a joint distribution $p(x_1, x_2, x_3, \dots, x_n)$ assuming the conditionals $p(x_i | x_{-i})$ are known, where $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$.

We derive the Gibbs sampler's update equation for the hidden variables from the joint distribution and arrive at:

$$p(z_i = j | z_{-i}, T, C, U, QT) \propto \frac{N_{-i,j}^{(t_i)} + \beta}{\sum_{u=1}^{N_d} N_{-i,j}^{(t_i)} + T\beta} \frac{N_{-i,j}^{(d_i)} + \alpha}{\sum_{k=1}^K N_{-i,k}^{(d_i)} + K\alpha} \frac{N_{-i,j}^{(u_i)} + \gamma}{\sum_{u=1}^{M_d} N_{-i,j}^{(u_i)} + U\gamma}$$

$$\prod_{u=1}^{M_d} \frac{M_{-i,j}^{(u_i)}}{M_i^{(u_i)}} \frac{N_{-i,j}^{(d_i)}}{N_d} \frac{N_{-i,j}^{(t_i)} + \eta}{\sum_{qt_l=1}^{L_d} N_{-i,j}^{(t_i)} + QT\eta} \prod_{qt_l=1}^{L_d} \frac{L_{-i,j}^{(qt_l)}}{L_u^{(qt_l)}} \frac{N_{-i,j}^{(d_i)}}{N_d}$$

where $N_{-i,(c)}^{(c)}$ is a number excluding the current position assignments of z_i , e.g., $N_{-i,j}^{(t_i)}$ is the number of tag t_i generated by the j th topic excluding the current position, $M_i^{(u_i)}$ is the total number of tags associated with URL u_i , and $M_{-i,j}^{(u_i)}$ is the number of tags associated with URL u_i that are assigned to topic z_j excluding the current position. Similarly, $L_u^{(q_{t_i})}$ is the total number of URLs that can be retrieved by a query including query term q_{t_i} and $L_{-i,j}^{(q_{t_i})}$ is the number of corresponding URLs that are assigned to topic z_j excluding the current position.

For any simple sample, we can estimate $\theta, \varphi, \psi, \pi$ using:

$$\theta_j^{(d_i)} = \frac{N_j^{(d_i)} + \alpha}{\sum_{k=1}^K N_k^{(d_i)} + K\alpha}, \quad \varphi_{j,t_i} = \frac{N_j^{(t_i)} + \beta}{\sum_{t_i=1}^{N_d} N_j^{(t_i)} + T\beta}, \quad \psi_{j,u_i} = \frac{N_j^{(u_i)} + \gamma}{\sum_{u_i=1}^{M_d} N_j^{(u_i)} + U\gamma}, \quad \pi_{j,q_{t_i}} = \frac{N_j^{(q_{t_i})} + \eta}{\sum_{q_{t_i}=1}^{L_d} N_j^{(q_{t_i})} + QT\eta}$$

3.3 Retrieval and Ranking of URLs

Based on the RTU-LDA model, we can compute the probability of each query being associated with each of the topics according to the probability of each term of the query being associated with the topic. Given a query q composed of N terms, i.e., $q = \{q_{t_1}, q_{t_2}, \dots, q_{t_n}\}$, the probability of q being associated with all K topics $T = \{T_1, T_2, \dots, T_k\}$ can be expressed as a $N \times K$ matrix:

$$p(q|T) = \begin{bmatrix} p(q_{t_1}|T_1) & p(q_{t_1}|T_2) & \dots & p(q_{t_1}|T_K) \\ p(q_{t_2}|T_1) & p(q_{t_2}|T_2) & \dots & p(q_{t_2}|T_K) \\ \dots & \dots & \dots & \dots \\ p(q_{t_n}|T_1) & p(q_{t_n}|T_2) & \dots & p(q_{t_n}|T_K) \end{bmatrix}$$

We then get the probability distribution of q over all K topics using the Bayes law:

$$[p(T_1|q) \quad p(T_2|q) \quad \dots \quad p(T_K|q)] \propto \left[\frac{\sum_{i=1}^n p(q_{t_i}|T_1)}{\sum_{k=1}^K \sum_{i=1}^n p(q_{t_i}|T_k)} \quad \frac{\sum_{i=1}^n p(q_{t_i}|T_2)}{\sum_{k=1}^K \sum_{i=1}^n p(q_{t_i}|T_k)} \quad \dots \quad \frac{\sum_{i=1}^n p(q_{t_i}|T_K)}{\sum_{k=1}^K \sum_{i=1}^n p(q_{t_i}|T_k)} \right]$$

Another query $q' = \{q_{t'_1}, q_{t'_2}, \dots, q_{t'_m}\}$ is related to q if they have similar probability distributions over the topics, which can be measured using the Kullback-Leibler

$$\text{Divergence: } D(q' \| q) = \sum_{k=1}^K p(T_k | q') \log \frac{p(T_k | q')}{p(T_k | q)} \quad (1)$$

Therefore we can reformulate an imprecise query with related existing queries through discovering the relevance between two queries based on their KL-divergence computed by formula (1). Queries are related if their divergence value is small, i.e., if their query terms have similar probabilities of being associated with the same topic.

The probability of URL u being associated with all the K topics is:

$$p(u|T) = [p(u|T_1) \quad p(u|T_2) \quad \dots \quad p(u|T_k)]$$

We also get the probability distribution of u over all K topics:

$$[p(T_1|u) \quad p(T_2|u) \quad \dots \quad p(T_k|u)] = \left[\frac{p(u|T_1)}{\sum_{k=1}^K p(u|T_k)} \quad \frac{p(u|T_2)}{\sum_{k=1}^K p(u|T_k)} \quad \dots \quad \frac{p(u|T_k)}{\sum_{k=1}^K p(u|T_k)} \right]$$

The relevance between q and u is measured by their similarity of probability distributions over the topics: $D(u||q) = \sum_{k=1}^K p(T_k|u) \log \frac{p(T_k|u)}{p(T_k|q)}$ (2)

If q 's terms and u have similar probabilities of being associated with all topics, URL u is likely to be relevant to query q .

We then use the weighted Borda count method [1] as a rank aggregation method to combine the URL list retrieved through the RTU-LDA model with the lists retrieved by a conventional search engine. A score is computed and assigned to each candidate URL according to the URL's position in each ranked lists and all candidate URLs are then ranked according to their total scores.

In summary, retrieval and ranking using our model are through the following steps:

- 1) When a user generates a new query, it will be reformulated with related existing queries discovered using (1). The terms of these queries have high probabilities of being associated with the topic that the terms of the new query are about.
- 2) The URLs associated with these queries will be discovered using formula (2).
- 3) Ranking scores will be computed and assigned to each of the URLs retrieved at step 2) as well as by a conventional search engine.
- 4) All URLs are ranked according to their scores. Top ranked URLs are those that have the highest probabilities of co-occurrence in the topic that the terms of the new query are about.

4 Experiments

We have conducted a set of experiments to validate that: 1) the proposed query reformulation approach based on the RTU-LDA topic model can retrieve more relevant results than alternative approaches, and 2) the retrieved results can be ranked according to their relevance to the goals of users' informational search tasks.

Our experimental data is derived from two real-world datasets: query logs from a commercial search engine AOL and a social annotation dataset from Delicious. The experiments compare our approach with the following alternative approaches:

- 1) *Baseline approach*: the approach used by the AOL search engine to retrieve and rank results. It is worth clarifying that because AOL's retrieval and ranking techniques are not public, the baseline approach is actually derived from logs.
- 2) *Query-clustering approach*: this approach uses a query-clustering algorithm similar to the one proposed in [2] to cluster semantically connected queries. The degree of similarity between two queries is decided by the fraction of common terms in the clicked URLs. The URLs returned by all the queries in a cluster are ordered according to the number of clicks occurring to them, which leads to the

popularity ranking of URLs. The new ranking of URLs is boosted by combing the popularity ranking and the original ranking returned by the search engine.

- 3) *Learning-to-rank approach*: for a sequence of queries generated by a user, this approach uses the algorithm developed in [10] to generate preference judgments about the relative relevance of the documents retrieved by an individual query and also those retrieved by different queries in the sequence. The ranked retrieval function is learned from the preference judgments using a ranking SVM.

4.1 Datasets

The AOL query logs used in the experiments contain about 20 million search queries from 657,426 users [9]. For each query, the URLs clicked by the user and the ranking of each clicked URL on the result list were recorded in the logs. We first removed the stop words and stemmed the query terms. For the purpose of preserving privacy and removing noises in the training process, we only kept those queries that contain no rare terms (i.e., terms occurred less than five times).

Because our objective is to compare the performance of the proposed probability-based approach with alternative ones in terms of supporting informational search tasks, we randomly chose 100 users from the dataset and manually extracted two informational search tasks accomplished by each user as the test dataset, i.e., 200 informational search tasks in total for the experiments. Each task is composed of a set of inter-related queries $\{q_1, q_2, \dots, q_n\}$ ($n > 1$) and a list of URLs $[u_1, u_2, \dots, u_m]$ ($m > 1$) retrieved by the set of queries and ranked according to their relevance to the goal of the user's informational search task (the relevance judgments were provided by the participants of these experiments). Well-defined metrics are used to evaluate the performance of each approach in accomplishing these tasks.

The remainder of the query logs were used for training the model and the algorithms. For the probability-based approach, a user's query terms and clicked URLs are treated as a document, which might be labeled with social annotations. The social annotation dataset used in the experiments is DeliciousT140 from Zubiaga [19], which was created with data retrieved from the social bookmarking site Delicious and various Web sites. This dataset contains 144,574 unique URLs and 67,104 different corresponding social tags retrieved from Delicious on June 2008. For a document containing a set of query terms and URLs, if a URL could be retrieved from this social annotation dataset, all its associated tags would be used as the labels of the document for training.

4.2 Methodology

The experiments were conducted to evaluate the performance of our approach and alternative approaches in terms of the model quality and the retrieval quality.

For the model quality, we compared the proposed model with the existing LDA and sLDA [4] models, in terms of the ability to identify latent topics for discovering the relevance between URLs and query terms. The LDA model is based on the co-occurrence probabilities of the query terms and URLs. The click-through data is

treated as a bag-of-words document, where “words” are query terms or URLs. The sLDA model selects one tag for each document as the label for supervised training. The most popular tag among all the tags of the document’ annotation is selected.

To evaluate the quality of a model, we measure its performance in discovering the relevance between the URLs and a query’s terms using the perplexity of held-out query-related URLs for the query. Perplexity is widely used in language modeling and can be defined as follows [19]:

$$Perplexity(D) = \exp\left\{-\frac{\sum_{q \in D} \sum_{u \in N_q} \ln P(u|q)}{\sum_{q \in D} N_q}\right\}, \text{ where } D \text{ is a}$$

test document, q is a query in the test document, N_q is the set of URLs that are related to q , and $P(u|q)$ is the probability of URL u being associated with query q inferred.

As perplexity is algebraically equivalent to the inverse of the geometric mean per-word likelihood, a low perplexity represents high performance. As LDA-based models are sensitive to the number of topics, we can optimize the models by analyzing the influence of perplexity using different topic numbers.

For the retrieval quality, we compared the retrieval results of our approach with those of the baseline, query-clustering, and learning-to-rank approaches, in terms of the ability to retrieve and rank URLs that are relevant to a search goal.

A number of methodologies and metrics have been used to evaluate the quality of retrieval. In these experiments, we specifically want to evaluate how the retrieved results have satisfied the goal of an informational search task in terms of search accuracy and search efficiency.

Search Accuracy. We measure the search accuracy using the following metrics:

(1) Relevance accuracy: given the search results of the test dataset, we measure how many relevant documents have been retrieved. It should be pointed out that the relevance is relative to the overall goal of an informational search task rather than to each individual query used in the search process. Evaluation metrics of Precision, Recall and MAP (Mean Average Precision) are used to measure relevance accuracy.

Precision is the fraction of retrieved documents that are relevant, while Recall is the fraction of relevant documents that have been retrieved.

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant} | \text{retrieved}), \text{ Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved} | \text{relevant})$$

Recall and Precision usually contradict each other. An informational search task would have a low Recall if retrieved results were only those that match some of the query terms. Our approach can retrieve more results that are relevant to a search goal, thus achieving high Recall, while keeping reasonable high Precision at the same time.

To reflect the overall performance of a search system, MAP provides a single-figure measure of quality across Recall levels. For a single information need, Average Precision is the average of precisions computed at the point of each of the relevant documents in the ranked sequence: $AP = \frac{1}{Rel} \sum_{r=1}^{Rel} \frac{Pos_r}{r}$, where Rel is the total number of

documents relevant to the query, and Pos_r is the position of the r^{th} relevant document

in the list of all resultant documents. MAP has been shown to have especially good discrimination and stability.

(2) Ranking accuracy: given the search results of the test dataset, we measure how accurately these results are ranked according to the relevance to the search goal. Again, the relevance is relative to the overall goal of an informational search task rather than to each individual query used in the search process.

Normalized Discounted Cumulative Gain (NDCG) is used to measure ranking accuracy. For every rank position k in the ranked list, NDCG is defined as follows:

$$NDCG(k) = \frac{1}{Z_k} \sum_{p=1}^k \frac{2^{\mathcal{S}(p)} - 1}{\log(1+p)},$$

where $\mathcal{S}(p)$ is the relevance score of the document at position p in the ranked list and Z_k is a normalization factor.

Search Efficiency. Search efficiency measures how effective the queries are. In particular, an informational search task would have a high search efficiency if the user generated few queries but each query retrieved many relevant documents.

The following SE (Search Efficiency) metric is defined to measure search efficiency: $SE = \frac{1}{N} \sum_{i=1}^N Run_k(q_i)$, where N is the total number of queries generated by

the user to achieve their search goal and $Run_k(q_i)$ is the number of relevant URLs in each query's top- k retrieved results. To obtain the statistical significance in measuring the search efficiency, we choose queries from the test dataset randomly in three different orders and calculate the average SE scores.

4.3 Results

Model quality. Fig 3 shows the perplexity tested on the held-out sample of each of the LDA-based models for different topic numbers $K = 20, 50, 100, 150, 200, 250, 300$. It is clear that for any fixed number of latent topics K , the RTU-LDA model has achieved the lowest perplexities, indicating that it is most capable to discover the latent relevance between URLs and query terms among the three LDA-based models.

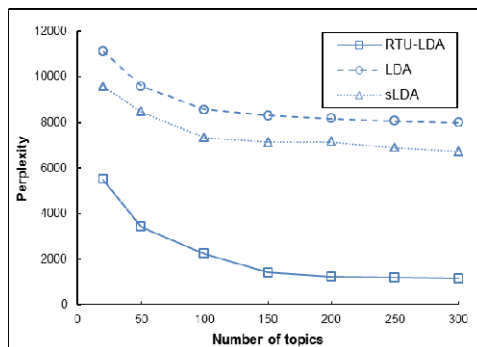


Fig. 3. The perplexities of different models for $K = 20, 50, 100, 150, 200, 250, 300$

The high perplexity of LDA is probably due to the polysemy of query terms and the ambiguity of queries in informational search tasks. For example, given the top ten words associated with a single topic learned by LDA: “home”, “speed”, “monitor”, “real”, “free”, “race”, “estate”, “window”, “furniture”, and “download”, LDA will probably infer that the queries of “home estate” and “speed up windows” are related. In contrast, RTU-LDA will derive one topic where “home” and “estate” are top words and another topic where “window” and “download” are top words.

Fig 3 also shows that the perplexity of RTU-LDA becomes steady when the number of topics reaches 200. Therefore we set $K=200$ as the number of topics for deriving latent topics and estimating parameters. To be consistent, we set the same model hyper parameters ($\alpha=50/K$, $\beta=0.1$) for all models in the experiments.

Precision and recall. Fig 4(a) shows the precision-recall graph of the four approaches. It can be seen that RTU-LDA-based approach outperforms the baseline approach in most cases except at the low recall level. Because our objective is to retrieve more documents that are relevant to the goal of an informational search task, we expect high precision at the high recall level. The baseline approach is not able to achieve high recalls due to the fact that the search engine has only retrieved the documents that match some of the query terms.

The query clustering approach has improved the recall through retrieving more documents that are relevant to the search goal. However, since it only clustered queries with common terms in the clicked URLs, documents with latent relevance to the queries could not be retrieved. Therefore this approach’s precisions are worse than RTU-LDA’s, either at low or high recall levels. The learning to rank approach is better than the baseline because it can include documents originally not present in the initial search results through learning. However, it is not easy to infer preference judgments from query logs. Compared to the RTU-LDA-based approach, this approach’s performance is worse especially at high recall levels, probably because the ranking SVM is more dependent on the proper preference judgments and the training data, while RTU-LDA is a generative probabilistic model and can deal with unseen queries more accurately.

Table 2 shows the MAP scores of the four approaches, which further illuminate that our approach has significantly improved the overall precision of informational search tasks, as compared to the query clustering and learn to rank approaches.

Fig 4(b) shows the NDCG@k ($k=1, \dots, 10$) of the four approaches. It is clear that RTU-LDA has achieved the best ranking accuracy in terms of the relevance of the retrieved URLs to the search goals, while the baseline is the worst and the other two approaches generally outperform the baseline. RTU-LDA uses the discovered latent semantic relationships between URLs and query terms to identify the URLs that are most relevant to the search goal. Ranking in the query clustering approach is only based on URL popularity, which cannot accurately reflect the relevance between URLs and query terms. The learning to rank approach, albeit using the rank SVM to learn ranked retrieval functions, mainly evaluates the ranking functions based on click

counts. Therefore both approaches tend to rank frequently clicked URLs higher, while RTU-LDA ranks the results based on the inferred latent relevance between URLs and query terms, thus achieving high ranking accuracy.

Search efficiency. Table 2 also shows the SE values of the four approaches. It can be seen that RTU-LDA has achieved a significant 36.8% improvement over the baseline, while the query clustering and learning to rank approaches have achieved 25.7% and 28.1% improvements over the baseline respectively. Generally speaking, compared to a baseline search engine, an RTU-LDA-based search engine requires significantly less number of queries - with each query retrieving significantly more relevant results - to achieve the goal of an informational search task. Therefore, the user can save a lot of time reformulating imprecise queries, clicking URLs and viewing Web pages that are irrelevant to their search goal at all.

Fig 5 shows the retrieved results of the query “windows safety & security” using the AOL and an RTU-LDA-based search engines respectively. It is clear that AOL only retrieved Web pages that matched some of the query terms, while RTU-LDA retrieved other Web pages that do not match any query term at all. More importantly, Web pages retrieved by RTU-LDA are all relevant to the search goal, while only the second one on AOL’s list is relevant to the search goal. RTU-LDA can infer search goals by discovering the latent semantic relationships between query terms and clicked URLs. In this example, because the Web pages retrieved by RTU-LDA are those which were annotated by the “computer” tag and whose URLs have been clicked most frequently by the users who generated such query terms as “windows”, “safety”, and “security”, RTU-LDA can derive a latent topic on “computer” to infer the search goal for “computer windows security”. If the search goal were something about “home window safety”, the query would be something like “home window safety” or “safe window film”. In that case, RTU-LDA would derive a different latent topic on “home” to infer that search goal. In conclusion, by retrieving and only retrieving Web pages that are directly relevant to search goals, RTU-LDA can achieve high search efficiency.

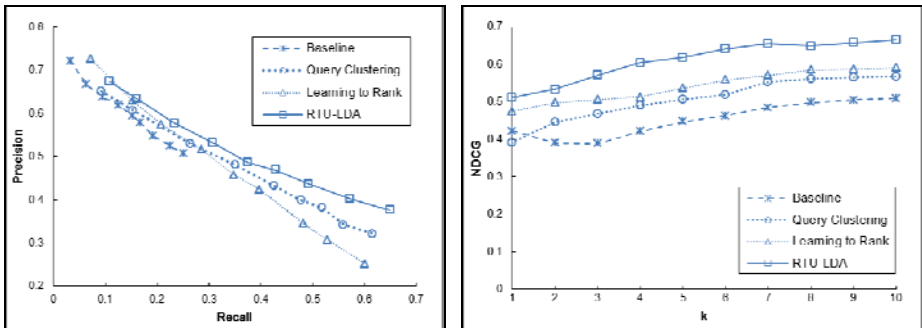


Fig. 4. (a) P-R graph of the four approaches (b) NDCG@k of the four approaches

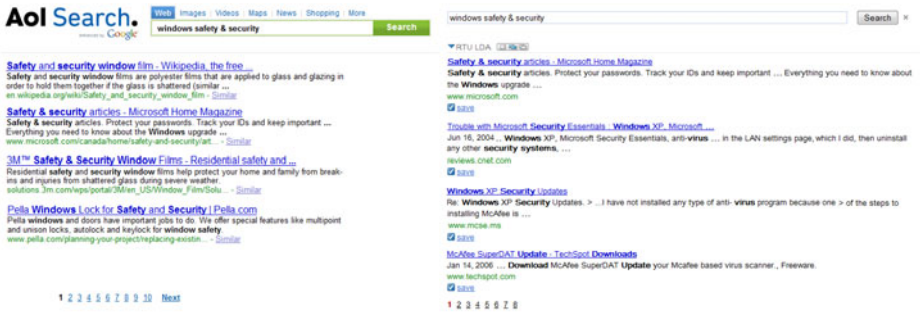


Fig. 5. Left: results from AOL; Right: results from RTU-LDA

Table 2. The MAP and SE (Search Efficiency) scores of the four approaches

Approaches	MAP	SE
Baseline	0.2204	2.53
Query Clustering	0.2536	3.18
Learning to Rank	0.2627	3.24
RTU-LDA	0.2779	3.46

5 Conclusions and Future Work

In this paper, we discussed the issues related to query reformulation in supporting informational search tasks. The query clustering approach to query reformulation can cluster queries based on their lexical similarity or common clicked Web page URLs. But many intrinsically related queries could not be clustered due to the sparse nature of the query space. Moreover, the retrieved URLs are not ranked according to their relevance to the goal of an informational search task.

The proposed approach is based on a novel RTU-LDA topic model to discover the latent semantic relationships between query terms and URLs. In this approach, not only can related queries be discovered based on the similarity of probability distributions over topics, but also the retrieved URLs can be ranked based on the similarity of probability distributions of the URLs and the queries. The performance of the proposed approach was evaluated by conducting experiments on real-world datasets. The results have shown that the model can accurately discover the latent semantic relationships between queries and URLs. Our approach thereby outperforms alternative ones such as the baseline, query clustering approach, and learning to rank approach in terms of both the search accuracy and the search efficiency.

We are currently investigating a further improvement of ranking accuracy by harnessing sentiment tags as the quality of Web pages can be indicated by these tags. We are also working on optimizing our topic model and the training algorithms as the time and space complexities are crucial for online Web search.

References

1. Aslam, J., Montague, M.: Models for metasearch. In: SIGIR 2001, pp. 276–284 (2001)
2. Baeza-Yates, R., Hurtado, C., Mendoza, M.: Query Clustering for Boosting Web Page Ranking. In: Favela, J., Menasalvas, E., Chávez, E. (eds.) AWIC 2004. LNCS (LNAI), vol. 3034, pp. 164–175. Springer, Heidelberg (2004)
3. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: KDD 2000, pp. 407–416 (2000)
4. Blei, D.M., McAuliffe, J.D.: Supervised topic models. In: NIPS 2007: Proceedings of Advances in Neural Information Processing Systems (2007)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *Journal of Machine Learning Research*, 993–1022 (2003)
6. Cui, H., Wen, J.R., Nie, J.Y., Ma, W.Y.: Query expansion by mining user logs. *IEEE Transaction of Knowledge Data Engineering* 15(4), 829–839 (2003)
7. Griffiths, T.L., Steyvers, M.: Finding scientific topics. *Proceeding of the National Academy of Sciences*, 5228–5235 (2004)
8. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD 2002, pp. 133–142 (2002)
9. Pass, G., Chowdhury, A., Torgeson, C.: A Picture of Search. In: *Infoscale 2006: Proceedings of the 1st International Conference on Scalable Information Systems* (2006)
10. Radlinski, F., Joachims, T.: Query Chains: Learning to Rank from Implicit Feedback. In: KDD 2005, pp. 239–248 (2005)
11. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) *The SMART Retrieval System*, pp. 313–323. Prentice Hall, Inc., Englewood Cliffs (1971)
12. Rose, D.E., Levinson, D.: Understanding user goals in web search. In: WWW 2004, pp. 13–19 (2004)
13. Wei, J., Bressan, S., Ooi, B.C.: Mining term association rules for automatic global query expansion: methodology and preliminary results. In: WISE 2000, pp. 366–373 (2000)
14. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: SIGIR 2006, pp. 178–185 (2006)
15. Wen, J.R., Nie, J.Y., Zhang, H.J.: Query clustering using content words and user feedback. In: SIGIR 2001, pp. 442–443 (2001)
16. Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., Li, H.: Context-aware ranking in web search. In: SIGIR 2010, pp. 451–458 (2010)
17. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR 1996, pp. 4–11 (1996)
18. Xue, G.R., Zeng, H.J., Chen, Z., Yu, Y., Ma, W.Y., Xi, W.S., Fan, W.G.: Optimizing web search using web click-through data. In: CIKM 2004, pp. 118–126 (2004)
19. Zhou, D., Bian, J., Zheng, S., Zha, H., Giles, C.L.: Exploring Social Annotations for Information Retrieval. In: WWW 2008, pp. 715–724 (2008)
20. Zubiaga, A., García-Plaza, A.P., Fresno, V., Martínez, R.: Content-based Clustering for Tag Cloud Visualization. In: ASONAM 2009 (2009)

A QoS-Aware Web Services Selection Model Using AND/OR Graph

Hong Yu and Man Liu

Institute of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, 400065, P.R. China
yuhong@cqupt.edu.cn

Abstract. With the increasingly emerging Web services, QoS-aware service selection is an active research area on Web services composition. It is a complex combinatorial optimization problem, which solves how to find a best composition plan that maximizes user's QoS requirement. In this paper a QoS-aware Web services selection model is proposed using AND/OR Graph after discussing the QoS criteria. The model is not only capable of dealing with sequence relations and fork relations, but also capable of dealing with parallel relations between services, and the multi-objective constraint function is defined to meet the QoS. Furthermore, a novel service selection algorithm is proposed based on the ant colony optimization. Finally, the algorithm is tested for the performance.

Keywords: Web Services, Quality of Service, AND/OR Graph, Ant Colony Optimization.

1 Introduction

Web services are defined as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. In service-oriented computing (SOC), Web services are fundamental elements of distributed and heterogeneous applications. With the development of theory and technology of Web Service, the users will demand more value added and informative services rather than those offered by single, isolated Web services. Therefore, Web services composition is a main method of implementing service reuse [1]. So developers and users can solve complex problems by composing available elementary services. Take an integrated financial management Web service as an example, it can be created by composing more specialized Web services for payroll, tax preparation, and cash management.

In addition, there are many different Web services available that could provide the same service function. However, these services have different Quality of Service(QoS) including execution price, execution duration, and availability etc. In order to satisfy the global QoS requirement of users, the best services must be selected from numerous candidates to compose a more complex service.

Obviously, the QoS specification and description are very important, and some researchers have studied on the QoS modeling. QoS criterion serves as a key

index for discriminating candidate Web services and Web service compositions with identical functionality. Cardoso etc. [2] propose the factors which QoS value model should have cost, time and reliability. Zeng etc. [17] describe the factors which QoS value model should have execution price, execution duration, reputation, successful execution rate and availability. In reference [17] the reputation is a real number, and Claro etc. [3] propose some improvement reputations based on fuzzy numbers. Hu etc. [6] describe the factors of QoS value as execution cost, execution time, availability and reliability. Zhang etc. [18] add the composability factor, and use the reliability factor to substitute the successful execution rate. Wan etc. [10] describe the factors of QoS value as execution price, execution duration, reliability and availability. Xiong etc. [14] describe the factors of QoS value as cost, availability, reliability, successful execution rate. Observing the different QoS criteria, we can see that the criteria mainly are about service price, execution during, availability and so on. Therefore, in this paper, we will mainly consider the five factors proposed by Zeng etc. [17].

The composite service is not a simple combination of many services. Relations between them will affect the quality and implementation of the composite service. Besides, not only the relations between services input and output parameters, but also the parallel relations or the fork relations between elementary services should be considered, too. For example, to rank and optimize composition, Lecue [8] uses the semantic similarities between output and input parameters of web service and other criteria such as quality of service. On the other hand, the fork relations or sequence relations are considered in literature [13] [5] [9] [11] [12] [16]. However, there are few literatures to mention how to deal with parallel relations even if parallel relations are discussed in different ways. It is easy to remind us thinking the parallel and fork relations as AND/OR relations. Thus, Lang and Su [7] present a formalization of the Web service composition problem as a search problem in an AND/OR graph, where the information provided in a service request are represented as an AND/OR graph, where AND/OR nodes are the services and the input/output documentary, respectively. However, the approach neglects the QoS and the human interference needed. Thus, we will propose a novel QoS-aware Web services composition model using AND/OR graph, where AND nodes and OR nodes mean the parallel relation and the fork relation between services, respectively.

As we have discussed, Web service selection problem is formulated as a multi-objective optimization problem. Ant colony optimization (ACO) [4], is a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems. The main idea of ACO is to model the problem as a search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behavior so that it will typically only find rather poor-quality paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony. ACO approach has been demonstrated by successful applications in a variety of hard combination optimization problems such as traveling salesman problems, vehicle routing problem, constraint satisfaction problem, machine learning, etc [15].

Inspired by the character of ant colony optimization, more researchers [13] [5] [9] [11] have focused on solving the service selection problem with ACO, which only consider the sequence relation between services. Wang etc. [12] take into account the fork relation between service. Zheng and Luo [19] present a QoS-aware Web services selection algorithm based on Ant Colony System, which is applied to Web services composition with selection and concurrent execution path, and present the concept of utility function of QoS which is the objective function of their algorithm, but the utility function is mono-objective problem with QoS constraints. Thus, we will describe a QoS-aware Web service composition selection algorithm using AND/OR graph based on ACO in this paper, which considers not only the sequence relation, but also the fork relation as well as the parallel relation between services, and define the multi-object function as the heuristic information.

The rest of this paper is structured as follows. First, we introduce some basic concepts about the quality criteria and the logical relation between services. A composite service selection model is proposed based on AND/OR graph in section 3, and the services selection problem is formed as a multi-objective optimization problem. Then a novel Web service composition selection algorithm using ACO is devised. The experiment results in Section 5 show that the novel algorithm is valid and reasonable. Some conclusions will be given in Section 6.

2 Basic Concepts

The quality criteria of elementary services and the logical relation between services will be discussed in this section.

2.1 Quality Criteria for Elementary Web Service

If a Web service includes one function(operation), it is called an elementary Web service, and Web service for short, an elementary Web services is formalized as a triple tuple as follows.

Definition 2.1 (*Web Service*). A Web service (WS) is defined as a tuple: $WS = (S, C, Q)$, where S denotes the basic information descriptions of the Web service, such as the name of Web service, commercial entities of Web service and textual descriptions of Web service; C denotes function descriptions of the Web service, including the service interface parameters, pre-conditions and post-conditions; Q denotes the factors of quality of the service.

Generally, Q is denoted as a array: $Q = (q_1, q_2, \dots, q_n)$, where q_i ($1 \leq i \leq n$) means quality criteria such as execution price of the Web service, execution duration, successful execution rate, reputation and availability, etc.

Web services interact through message, which is defined as the interface parameter of Web services [17].

Definition 2.2 (*Interface Parameters of Web Services*). The interface parameter of Web services is defined as a binary group $Parameter=(I, O)$.

Here, I is the set of input parameters, $I = \{i_1, i_2, \dots, i_n\}$; O is the set of output parameters, $O = \{o_1, o_2, \dots, o_m\}$. The elements of I and O are described as the triple: (M, T, U) , where M denotes the meaning of parameter, T denotes the type of parameter, U denotes the unit parameter.

Generally speaking, an elementary service includes many physical services (PS for short), each of the physical services can achieve the same function by itself. In other words, there are many service providers can provide the same commercial service, which is also called the physical service, and the different physical services have different QoS property values. The goal of Web service selecting is to choose elementary service, and physics services is bound when executing the services composition.

In this paper, the five generic quality criteria [17] for elementary services (WS) will be considered, which are execution price ($q_{pr}(WS, op)$), execution duration ($q_{du}(WS, op)$), reputation ($q_{rep}(WS)$), successful execution rate ($q_{rat}(WS)$) and availability ($q_{av}(WS)$).

Thus, the Quality of Service (QoS) criterion for an elementary service WS is defined as $Q(WS, op) = (q_{pr}(WS, op), q_{du}(WS, op), q_{rep}(WS), q_{rat}(WS), q_{av}(WS))$.

2.2 Meta-control Logical Relation Between Services

A composite service combine some elementary Web services according to the logic to achieve the service requirement. In order to express convenient, two virtual elementary services are given as the start point and the end point of the composite service, expressed as WS_0 and WS_{n+1} , respectively.

Definition 2.3 (*Composite Web Service*). A composite Web service (CWS) is defined as a tuple: $CWS = (WS, CR)$.

Here WS is called a composite service, which is a set of elementary services. WS_i is an elementary Web service, namely $WS = (WS_0, WS_1, \dots, WS_i, \dots, WS_n, WS_{n+1})$. Elementary services and composite service are Web services as well, the difference is merely on the size of the granularity.

CR is a set of the control logical relation between services. For two components WS_i and WS_j , there exists one of the four following relations between them, which is called meta-control logical relation between WS_i and WS_j . The relation also is described in Figure 1.

- **Sequence Relation:** For two component services WS_i and WS_j , if WS_j is executed after WS_i , we call that there is a sequence relation between WS_i and WS_j , expressed as $ControlRelation(WS_i \cdot WS_j)$, and shown in Part(1) of Figure 1.
- **Iteration Relation:** For a component services WS_i , if WS_i is executed for many times, we call that there is an iteration relation of WS_i , expressed as $ControlRelation(!WS_i)$, and shown in Part(2) of Figure 1.
- **Parallel Relation:** For two component services WS_i and WS_j , which have the same foregoing service WS_s , if both of them need to be executed, we

call that there is a parallel relation between WS_i and WS_j , expressed as $ControlRelation(WS_i \wedge WS_j)$, and shown in Part(3) of Figure 1.

- **Fork Relation:** For two component services WS_i and WS_j , which have the same foregoing service WS_s , if only one of them need to be executed, we call that there is a fork relation between WS_i and WS_j , expressed as $ControlRelation(WS_i \vee WS_j)$, and shown in Part(4) of Figure 1.

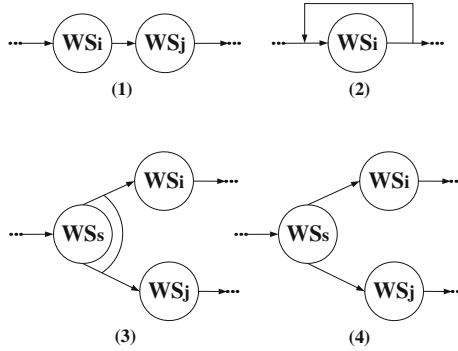


Fig. 1. Meta-control Logic Relation Between Services

Obviously, the iteration relation is a special case of the sequence relation when the pre-service and the post-service are the same one. Therefore, we just need to discuss the other three meta-controlled logical relations from now on.

3 Composite Service Model Using AND/OR Graph

In this section, we will describe the composite service model based on the AND/OR graph and formalize the composite service selection problem.

3.1 AND/OR Graph for Composite Services

There are some solutions to deal with parallel relations or fork relations, but there are few solutions to dispose both parallel relations and fork relations in the current composite service models.

As we have seen in Figure 1, there is a logical AND relationship among the parallel relation services, and there is a logical OR relationship among the fork relation services. Thus, we decide to use AND/OR graph to form the composite services model. That is, the parallel relation is expressed by “AND” logic, and the fork relation is expressed by “OR” logic.

Obviously, we can represent a composite service as an AND/OR Graph, where a vertex is a Web service WS_i , and the direction of an edge means the data logical relation between the two adjacent vertices. Generally speaking, a vertex is called an AND vertex if whose succeeding vertices are parallel to each other,

and a vertex is called an OR vertex if whose succeeding vertices are fork to each other.

In other hand, there are more than one meta-control logic relation in most cases. Considering a mixed relationship, that is, there are parallel relation and the fork relation among the successors of a service vertex, we can transfer the mixed relationship to meta-control logic relations. Then we just need to discuss the model including meta-control logic relations.

Let us take Figure 2 as an example to explain the transfer method. In the part (1) of Figure 2, WS_j and WS_p are AND logical vertices, but WS_q is OR logical with WS_p or WS_j . We just need to add a new vertex, WS_s , as the previous point of the AND vertices. Then, there just leave the meta-control logical relations in the part (2) of Figure 2.

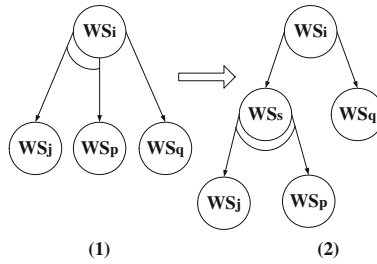


Fig. 2. The Transfer Method for Mixed Logical Relation

Here is an example to illustrate a Web composite service system structured by an AND/OR graph in Figure 3. In the example, the combination of Web Service (CWS) composes of 20 elementary services, two virtual services denoting the start and the end as WS_0 and WS_{21} , respectively. That is $CWS = \{WS_0, WS_1, \dots, WS_{20}, WS_{21}\}$. There are lots of choices from the start elementary service to the end elementary service. In addition, there are some logical relationships among the elementary services. Some are AND services, and some are OR services.

From Figure 3, we can see that there are one meta-control logic relation between arbitrary component services WS_i and WS_j as well as the data logic relation denoted by the directed arrows. For example, there are three choices from WS_1 to WS_2 , WS_3 and WS_4 , and they are the OR logic to each other. Both WS_5 and WS_8 must be executed after WS_4 , and they are the AND logic to each other.

On the other hand, there are usually more than one physical services for a web service WS_i . Namely, $WS_i = \{PS_{i1}, PS_{i2}, \dots, PS_{i|PS_i|}\}$, where $|PS_i|$ denotes the number of the physical service for WS_i . Figure 4 is a schematic diagram of physical services, which is based on the edges from elementary services WS_1 to WS_2 as an example in Figure 3. Suppose there are m physical services for each elementary service. There exist paths from each physical service of WS_1 to all physical services of WS_2 , formed a complete directed bipartite graph.

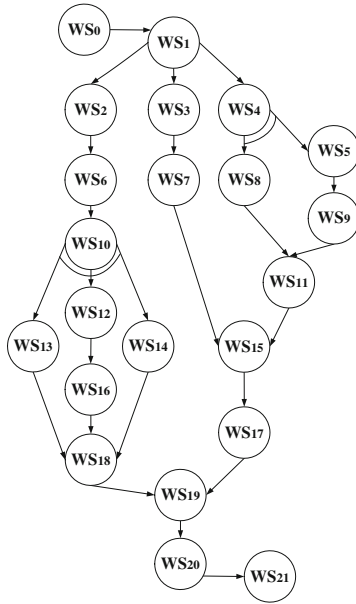


Fig. 3. A Composite Service System Based on AND/OR Graph

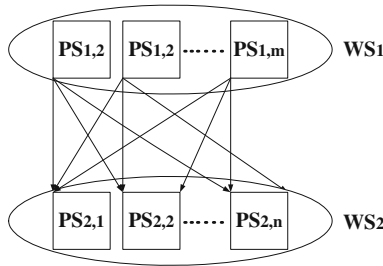


Fig. 4. Schematic Diagram of Physical Services

3.2 Description of Composite Service Selection Model

Given $G(N, E, W)$ is an AND/OR graph, where N is a set of the vertex, E is a set of the edge, W is a set of the edge weight. The mathematical model of multi-objective services composition optimization with QoS constraints is described as follows.

Objective functions:

$$\begin{cases} \min(q_{pr}(P)) \\ \min(q_{du}(P)) \\ \max(q_{rep}(P)) \\ \max(q_{rat}(P)) \\ \max(q_{av}(P)) \end{cases} \quad (1)$$

Constraint conditions:

$$\begin{cases} q_{pr}(P) < Pr \\ q_{du}(P) < Du \\ q_{rep}(P) > Rep \\ q_{rat}(P) > Rat \\ q_{av}(P) > Av \end{cases} \quad (2)$$

Where Pr , Du , Rep , Rat and Av denote constant values of the execution price, the execution duration, the reputation, the successful execution rate and the availability, respectively.

The execution duration and the execution price used are negative criteria, i.e., the higher the value is, the lower the quality is. And the reputation, the availability and the reliability are positive criteria, i.e., the higher the value is, the higher the quality is. There is a cumulative relationship between the negative criteria, and there is a multiplication between the positive criteria.

Normalize the multi-objection in equation (1), and we distribute the QoS values to the edges of the directed graph. That is, the goal of searching is to find a minimal weight path. If there are two edges with the same weight, choose the first edge. As we have discussed, the five factors are divided into the negative criteria and the positive criteria. In other words, the higher the prices and the execution durations are, the longer the distance is. On the contrary, the higher the positive criteria, the shorter the distance is. Thus, we can define the distance(weight) of edge $\langle i, j \rangle \in E$ as the follows.

$$d_{ij} = \begin{cases} \frac{Rep \times Rat \times Av \times (pr_i + pr_j) \times (du_i + du_j)}{Pr \times Du \times rep_i \times rep_j \times rat_i \times rat_j \times av_i \times av_j}, & \langle i, j \rangle \in E \\ \infty, & else \end{cases} \quad (3)$$

4 Web Service Composition Selection Algorithm Based on ACO

Literature [18], [16] and [19] consider both the parallel relation and fork relation between services in the composite services, but the problem of composite service selection is transferred to a mono-objective optimized function with QoS constraints. Literatures [13], [5], [9], [11], [12] solve the composite service selection problem with ACO, however they just consider the sequence relation between services.

Thus, we will propose a QoS-aware Web service composition selection algorithm using AND/OR graph based on ACO in this section, QCS-AND/OR-ACO algorithm for short. The algorithm considers not only the sequence relation, but also the fork relation as well as the parallel relation between services. In addition, the services selection is formed as a multi-objective optimization problem as discussed in the last section.

We set ants in the start service vertex in the composite service based on AND/OR graph. The task of each ant is to find a path, which satisfies the

multi-objective composition optimization functions and the QoS constraint conditions, from the start vertex to the end vertex. If ants arrive at an OR vertex, they visit one of the next service vertices according to the transition rules, and if ants arrive at an AND vertex, they visit all the succeeding service vertices. Every ant update the local pheromone when it passes an edge. After the colony complete an iteration, the global pheromone is updated. The algorithm is described in Algorithm 1.

The Ant Colony System differs from the previous ant system because of the three main aspects[4]. The key of the QCS-AND/OR-ACO algorithm is the state transition rule, the global updating rule and the local updating rule.

Transition Rule. Let P_{ij}^k denotes the state transition probability, that is, the ant k moves from service i to service j in the t times iteration.

q is a random variable, q_0 is a constant, $q \in [0, 1]$, and $q_0(0 \leq q_0 \leq 1)$. $allowed_k$ is the set of the nodes (physical services) allowed to visit by the ant k . When an ant k arrives at an OR vertex, $allowed_k$ is composed of the physical services in the all succeed vertices(services), and the ant chooses one node to visit according to the following transition rule. When an ant arrives at an AND vertex, the $allowed_k$ is the union of the composed of the physical services in every succeed vertices(services), the ant must choose one node to visit from every succeed vertices. The transition rule is given as follows.

$$P_{ij}^k = \begin{cases} \arg \max_{l \in allowed_k} [\tau_{il}]^\alpha [\eta_{il}]^\beta & \text{if } q < q_0 \\ J & \text{otherwise} \end{cases} \quad (4)$$

where J , a random variable, is computed by the following probability distribution equation.

$$J = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in allowed_k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & l \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$\eta_{ij} = \frac{1}{d_{ij}}$ is the heuristic information, d_{ij} is computed according to Equation (3). τ_{ij} is the pheromone intensity of the path between service i and j , α is a parameter controlling the importance of the pheromone, and β is a parameter controlling the importance of heuristic information.

Local Pheromone Updating Rule. The ant updates the pheromone on the passing edges according to the following equation.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (6)$$

where τ_0 is the initial pheromone level, ρ ($0 < \rho < 1$) is the pheromone decay parameter.

Algorithm 1. QoS-aware Web Service Composition Selection Algorithm
Using AND/OR Graph Based on ACO

Input: $G = (N, E, W)$.

Output: a service path GS .

begin

Step 1: Initially

$\tau_{ij}(0) = \text{const}$; $t = 0$;

$LS_k = \emptyset$; //local solution

$TS_k = \emptyset$; //temp solution

$GS_t, GS'_t = \emptyset$; //global solution

Set M, T ; //the number of ants, and the number of maximum iteration

Step 2: Construct solution

while $GS = GS'_t$ or $t = T$ **do**

for $k = 1, k \leq M, k++$ **do**

Set ants on the starting vertex WS_0 ; $currentnode = WS_0$;

while $currentnode! = \emptyset$ **do**

Update $allowed_k$;

if $currentnode$ is an OR vertex **then**

According to Equation (4) to choose PS_{ij} from all candidate services;

$LS_k = LS_k \cup PS_{ij}$;

According to Equation (6) to update the pheromone on the passing edges;

$currentnode = currentnode \cup PS_{ij}$;

end

else

for $\forall WS \in allowed_k$ **do**

According to Equation (4) to choose PS_{ij} from all candidate services;

$LS_k = LS_k \cup PS_{ij}$;

$currentnode = currentnode \cup PS_{ij}$;

end

According to Equation (6) to update the pheromone on the passing edges;

end

end

if LS_k is superior to TS_k **then**

$TS_k = LS_k$;

end

$LS_k = \emptyset$;

end

choose the best one from all of TS_k to GS'_t ;

$TS_k = \emptyset$;

According to Equation (7) to update the pheromone;

$t++$;

if GS'_t is superior to GS **then**

$GS = GS'_t$;

end

end

Step 3: Output the path GS .

end

Global Pheromone Updating Rule. After an iteration, the pheromone updated by the following equation.

$$\begin{aligned}\tau_{ij} &= (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij} \\ \Delta\tau_{ij} &= \sum_{k=1}^m \Delta\tau_{ij}^k \\ \Delta\tau_{ij}^k &= \begin{cases} \frac{1}{L_k} & (i, j) \in L_k \\ 0 & \text{otherwise} \end{cases}\end{aligned}\quad (7)$$

where L_k denotes the length of optimal execution path in this iteration.

5 Experiments and Analysis

5.1 Example Analysis

The advantage of the proposed method is processing both sequence relations and fork relations as well as parallel relations.

Let us take the composite service system in Figure 3 as an example again, to explain how the QCS-AND/OR-ACO algorithm to find a path which satisfy the multi-objective optimal functions and the constraint conditions.

For the virtual services WS_0 and WS_{21} , we set the five factors of QoS to be 0, 0, 1, 1 and 1 for execution price, execution duration, reputation, successful execution rate and availability, respectively.

Let $PS_{i,j}$ denotes the j -th physical service of the i -th elementary service, and $PS_{i,j}(A, B, C, D, E)$ means that A, B, C, D, E are the five QoS factors of $PS_{i,j}$ as execution price, execution duration, reputation, successful execution rate and availability, respectively. And set the ultimate values of the constrain conditions as $Rep = 4$, $Rat = 10$, $Av = 10$, $Pr = 5$, $Du = 4$.

Suppose we have the following QoS values:

$PS_{0,0}(0, 0, 1, 1, 1)$, $PS_{1,1}(8, 9, 8, 8, 9)$, $PS_{4,1}(2, 3, 4, 5, 6)$, $PS_{5,0}(3, 2, 1, 8, 9)$, $PS_{9,0}(3, 5, 4, 8, 8)$, $PS_{8,1}(1, 1, 8, 9, 8)$, and etc.

For example, when the ant visits the WS_4 , it steps on the 1-th physical service now, then it choose the 0-th physical service of the WS_5 according to Equation (4). Actually, the weight of the edge, $d[4, 1][5, 0]$, be calculated by Equation (3). That is, $d[4, 1][5, 0] = Rat \times Av \times Rep \times (du_{4,1} + du_{5,0}) \times (pr_{4,1} + pr_{5,0}) / Pr \times Du \times rep_{4,1} \times rep_{5,0} \times rat_{4,1} \times rat_{5,0} \times av_{4,1} \times av_{5,0} = 0.0578703$.

In the example, we can obtain an optimal composite service path, which is confirmed by the brute force searching.

5.2 Experimental Results

In order to evaluate the approach, the QCS-AND/OR-ACO algorithm is programmed in Visual C++ 6.0 and running in a PC, which is 2.60GHz and 1.96G Memory.

There are 4 services instance system, denoted by SI1, SI2, SI3, SI4. The number of elementary services in the composite service system SI1, SI2, SI3 and SI4

are 10, 20, 30 and 40, respectively. The number of candidate physical services for each elementary service can be different in practice. We set the number is 20 simply in the experiments. The service execution price and execution duration is a integral data which is generated randomly between 1 and 100 respectively, reputation is a integral data which is generated randomly which is from 1 to 10, successful execution rate and availability of a Web service is a float which is generated randomly between 0 and 1. Here, $\alpha = 2.0$, $\beta = 5.0$, $\rho = 0.1$, $q_0 = 0.9$; $Rep = 2$, $Rat = 0.6$, $Av = 0.5$, $Pr = 300$, $Du = 100$. Each experiment is tested 10 times, and the average results are calculated as the result of the experiment.

Experiment 1. In this paper, a QoS-aware Web services selection model is proposed using AND/OR Graph. The model is capable of dealing with sequence relations and fork relations, as well as parallel relations between services. In order to show that the new method is valuable, the approach in literature [12] is used here, where only fork relations and sequence relations between services are considered. The results are recorded in Figure 5 and Table 1, where CPU running time in seconds which start from ants start to search a path, not considering the time of data processing, *Ite* and *Dist* means the number of iterations and the distance of the path, respectively.

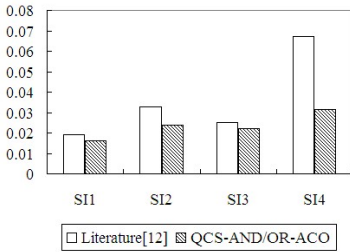


Fig. 5. Comparison of the CPU Time

Table 1. Comparison of the Distance

SI	Literature[12]		QCS-AND/OR-ACO	
	<i>Ite</i>	<i>Dist</i>	<i>Ite</i>	<i>Dist</i>
SI1	3.6	1.598631	2.0	0.039413
SI2	4.7	1.525001	2.8	0.093870
SI3	2.8	2.459177	2.3	0.065158
SI4	6.1	2.479946	2.9	0.054923

From Figure 5, we know that the CPU running time of QCS-AND/OR-ACO algorithm is more high-efficiency than the algorithm in Literature [12]. From Table 1, we know that the distance of the best optimal path obtained by QCS-AND/OR-ACO algorithm is shorter than the algorithm in Literature [12]. That is, to process sequence relations and fork relations, the new algorithm is better than the algorithm in Literature [12].

Experiment 2. In order to show that the new method is capable to deal with the parallel relations in the web composite service system, more experiments are tested. That is, we change some vertices in different proportions to be AND vertices, namely to be parallel relations, in the service instance system, SI4, which has 40×20 vertices. Then, the QCS-AND/OR-ACO algorithm is tested on the changed service instances. The results of experiment are in Table 2.

From Table 2, we can see that the new algorithm is not only deal with the sequence relations and fork relations, but also good at dealing with the parallel

Table 2. Research on the Different Proportion of AND Vertices in SI4

Proportion of AND vertices	CPU time	Distance of the best optimal path
0%	0.0313	0.054923
2.5%	0.0342	0.054078
5%	0.0404	0.054177
10%	0.0437	0.058206
20%	0.0466	0.028575

relations. Furthermore, with the increasing of the proportion of the AND nodes, the CPU running time and the distances are reasonable.

6 Conclusion and Future Work

QoS-aware service selection is a complex combinatorial optimization problem, which solves how to find a best composition plan that maximizes user's QoS requirement. Firstly, this paper presents the QoS criteria for composite services after discusse relations between elementary services. Secondly, a QoS-aware Web services selection model is proposed using AND/OR graph, which considers not only sequence relations, but also parallel relations as well as fork relations between elementary services. Then, the composite services selection model is formed and the muti-objection functions defined. Finally, this paper proposes a QoS-aware Web service composition selection algorithm using AND/OR graph based on ACO, and the distance between physical services is used as the heuristic information. Experimental results show that the novel algorithm is valuable. How to improve the efficiency of the algorithm is our further work.

Acknowledgment. This work was supported in part by the China NNSF grant 60773113 and the Natural Science Foundation of Chongqing of China grant CSTC2009BB2082.

References

1. Benatallah, B., Sheng, Q.Z.: The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing* 7, 40–48 (2003)
2. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of service for workflows and Web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web* 1, 281–308 (2004)
3. Claro, D.B., Albers, P., Hao, J.K.: Selecting Web Services for Optimal Composition. In: *IEEE International Conference on Web Services, Workshop on Semantic and Dynamic Web Processes*, pp. 32–45. IEEE Press, Orlando (2005)
4. Dorigo, M., Sttzle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)

5. Fang, Q., Peng, X., Liu, Q., Hu, Y.: A Global QoS Optimizing Web Services Selection Algorithm based on MOACO for Dynamic Web Service Composition. In: IEEE International Forum on Information Technology and Applications, pp. 37–42. IEEE Computer Society, Chengdu (2009)
6. Hu, J.Q., Guo, C.G., Wang, H.M., Zou, P.: Quality Driven Web Services Selection. In: 2005 IEEE International Conference on E-Business Engineering, pp. 681–688. IEEE Computer Society, Beijing (2005)
7. Lang, Q.-H.A., Su, S.Y.-W.: AND/OR graph and search algorithm for discovering composite Web services. *International Journal of Web Services Research* 2, 48–67 (2005)
8. Lécué, F.: Optimizing QoS-Aware Semantic Web Service Composition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 375–391. Springer, Heidelberg (2009)
9. Liu, Z., Wang, Z., Zhou, X., Lou, Y., Shang, L.: A New Algorithm for QoS-aware Composite Web Services Selection. In: 2nd International Workshop on Intelligent Systems and Applications, pp. 1–4. IEEE Press, Wuhan (2010)
10. Wan, C.L., Ullrich, C., Chen, L.M., Huang, R., Luo, J., Shi, Z.: On Solving QoS-Aware Service Selection Problem with Service Composition. In: Seventh International Conference on Grid and Cooperative Computing, pp. 467–474. IEEE Computer Society, Shenzhen (2008)
11. Wang, R.X., Ma, L.: The research of web service selection based on the ant colony algorithm. In: 2010 International Conference on Artificial Intelligence and Computational Intelligence, pp. 551–555. Springer, Sanya (2010)
12. Wang, Y., Dai, G.P., Jiang, Z.T., Hou, Y.R., Fang, J., Ren, X.T.: A Trust Enhanced Service Composition Scheduling Algorithm. *Acta Electronica Sinica* 37, 2234–2238 (2009) (in Chinese)
13. Wang, Y.W.: Application of Chaos Ant colony algorithm in web service composition based on QoS. In: 2009 International Forum on Information Technology and Applications, pp. 225–227. IEEE Press, Chengdu (2009)
14. Xiong, P., Fan, Y., Zhou, M.: Web Service Configuration Under Multiple Quality-of-Service Attributes. *IEEE Transaction on Automation Science and Engineering* 6, 311–321 (2009)
15. Yu, H., Wang, G., Lan, F.: Solving the Attribute Reduction Problem with Ant Colony Optimization. In: Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) Transactions on Rough Sets XIII. LNCS, vol. 6499, pp. 240–259. Springer, Heidelberg (2011)
16. Yu, T., Zhang, Y., Lin, K.J.: Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints. *ACM Transactions on the Web* 1, 6–32 (2007)
17. Zeng, L.Z., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for Web services composition. *IEEE Transactions on Software Engineering* 30, 311–327 (2004)
18. Zhang, G., Zhang, H., Wang, Z.: A QoS-based Web services selection method for dynamic Web service composition. In: First International Workshop on Education Technology and Computer Science, pp. 832–835. IEEE Computer Society, Wuhan (2009)
19. Zheng, X., Luo, J.: Ant Colony System Based Algorithm for QoS-Aware Web Service Selection. In: The 4th International Conference on Grid Service Engineering and Management GSEM 2007, pp. 39–50 (2007)

A Tweet-Centric Approach for Topic-Specific Author Ranking in Micro-Blog

Shoubin Kong and Ling Feng

Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
{kongsb09@mails., lingfeng@}tsinghua.edu.cn

Abstract. Most users play two roles in micro-blog, namely, *author* and *reader* of tweets. Facing diverse users and mass user-generated contents in micro-blog, identifying and ranking influential authors who post topic-specific high-quality contents is a challenge. In this paper, we present a way to measure the quality of tweets, which accordingly determines the influence of their authors. The quality of the tweet is evaluated according to the topic focus degree, the retweeting behavior, and the topic-specific influence of the users who retweet it. In this way, the relationships between two micro-blog users extend beyond the traditional *following* (i.e., *friend-follower*) relationship to have more that are established indirectly and dynamically through tweets. We explore the use of these enriched relationships and present a tweet-centric topic-specific author ranking in micro-blog. To enable timely mass data processing on a daily or even hourly basis, we implement our ranking method using MapReduce framework. Some evaluation experiments have been conducted based on a large-scaled real dataset from Tencent micro-blog, which has the largest number of users (over 200 millions) in China. The result shows that our author ranking approach outperforms the PageRank-based and HITS-based approaches significantly in terms of ranking accuracy and quality.

Keywords: Micro-blog, topic-specific, tweet, reader, author, ranking.

1 Introduction

1.1 Motivation

Micro-blog, a novel social network service, has spread rapidly over the world and been accepted by hundreds of millions of people. In micro-blog, the contents posted by users are no longer called blogs but *tweets*, which are short texts within the limit of 140 characters, and the contents can be anything that the users want to share, such as the state of their mind, what they are doing or seeing, hot news, funny jokes or videos, and so on. Besides, people can follow any other users whom they are interested in, and share or forward their contents without their permission. Such contents sharing or forwarding activity is also called *retweeting*,

which is the most pervasive user behavior in micro-blog nowadays. Since micro-blog is much more open than some other social network services like facebook, the speed of information diffusion among micro-blog users is incredible, and users' interaction becomes unprecedentedly easy and unimpeded.

Users in micro-blog are diverse, including organizations, companies, news media, celebrities, domain experts, individual users, and so on. Most of the users play two roles, *author* and *reader* of tweets. There is a demand to identify influential authors who post high-quality topic-specific tweets from a variety of micro-blog users. Consider the following scenarios.

- 1) *For an NBA fan, besides obtaining the game information from the sports channel in micro-blog, s/he may be more interested in the opinions written by authoritative authors on the NBA games, and want to interact with them further.*
- 2) *For one who wants to have an overview picture of Libya Crisis, news media in micro-blog may be insufficient. A war correspondent who posted original tweets about live situations of Libya may be more attractive*
- 3) *A policeman may want to trace influential authors of certain tweets related to a crime in micro-blog to get some clues for case detection.*

The aim of this study is to investigate techniques for topic-specific author recognizing and ranking in micro-blog.

So far, there are some work with respect to finding influential users in micro-blog, where authors and readers are uniformly treated as users without distinction, and users ranking is largely based on the links between users according to the *following* relationship (i.e., *friend* and *follower* relationship) [9]. However, the *following* relationship may not be the most important factor for recognizing topic-specific influential users due to the following four reasons. First, some following relationships may be just because users know each other, not because they have common interests in topics. Second, the *following* relationship does not distinguish the closeness degree of interest between two users on for a specific topic. Third, the following relationship does not reflect the difference of user behaviors, such as whether a follower retweets his/her friend's tweet either with or without comments; how much interest a user has in a topic. Fourth, the most important reason is that the establishment of the following relationships often lags behind new topics. For instance, when a war or an earthquake broke out, an eyewitness (an ordinary individual micro-blog user with few followers) may post many valuable tweets about the sudden event. In this case, the diffusion of valuable information through multi-hop retweeting is much faster than the establishment of the following relationships. Thus, relying on the following relationships is inadequate for user ranking [8].

1.2 Our Work

To overcome the deficiencies, we propose a tweet-centric approach for topic-specific author recognizing and ranking in micro-blog. The main idea is to capture more inherent relationships in the triangle of *author user*, *tweet*, and *reader*

user, and employ these rich relationships beyond users' static *following* relationship in authors' ranking. To this end, we build a weighted directed user-tweet model consisting of two types of nodes, namely, *user* node and *tweet* node. For a tweet, we distinguish user's role into author or reader explicitly. The edge from a *user* node to a *tweet* node means the user retweets the tweet (as reader), while that from a *tweet* node to a *user* node means the tweet belongs to the user (as author). A weight is bounded with each edge; the weight on user-to-tweet edges reflect whether the reader retweets the tweet with or without comments, while that on tweet-to-user edges distinguish the contributions of different-quality tweets to the influence of authors. In this way, users' relationships are dynamically established through real user-generated tweets and users' reaction behaviors, rather than directly through the static following relationships.

In the model, tweet score, author score, and reader score are defined to quantitatively measure the *quality* of a tweet, the *influence* of an author, and the *topic focus* of a reader, respectively. 1) The topic focus score of a user as reader (reader score) is the statistic percentage of topic-specific tweets s/he has retweeted. 2) The influence score of a user as author (author score) is determined by the quality of his/her originally posted tweets. 3) The quality score of a tweet (tweet score) is subject to the retweeting behaviors of its readers (i.e., with or without comments), as well as the interest and influence scores of its linked users (which may also be readers or authors of other tweets). This is based on two observations. First, a tweet retweeted by readers with comments intuitively attracts more attentions than those without comments, since people will not comment a tweet unless it is interesting enough. Second, a tweet retweeted (forwarded) by other influential authors or interested readers on the topic appears to have better quality than those retweeted by less influential and interested users.

Computation of the scores of tweets and users are iterative until all user scores are converged. Then users are ranked according to their influential scores as authors. We implement our ranking method using the MapReduce framework, ensuring mass micro-blog data processing within acceptable time, for example, on a daily or even hourly basis. We have conducted some experiments on a real large Micro-blog data from Tencent, which owns the largest number of users in China (over 200 millions). The experimental result shows that our author ranking approach outperforms the PageRank-based and HITS-based approaches in terms of ranking accuracy and quality.

In summary, the contributions of the study lie in the following two aspects.

1. We propose a tweet-centric approach to rank topic-specific influential authors in micro-blog. Different from the existing user ranking methods based on the simple following relationships between users, our approach captures more inherent relationships among tweets, authors, and readers in micro-blog, and explores the use of these rich dynamic relationships in ranking authors specifically.
2. We implement our author ranking algorithm using MapReduce framework, enabling timely identification of influential authors in the mass micro-blog. We evaluate the approach on the real large micro-blog data from Tencent,

and analyze the daily author ranking results on different kinds of topics. Our experimental result shows that our author ranking method outperforms the PageRank-based and HITS-based approaches in terms of ranking accuracy and quality.

The remainder of this paper is organized as follows. In Section 2, we review some closely related work and highlight the differences of our work from the existing ones. In Section 3, we describe our user-tweet interaction model. Details of the topic-specific author ranking algorithm are given in Section 4. We describe our performance study in Section 5, and conclude the paper in Section 6.

2 Related Work

Some work on user ranking in micro-blog came out in the last few years. TunkRank [7] and TwitterRank [9], which are both variants of PageRank [4], measured the influence of users in Twitter based on a user graph constructed according to following relationships. The difference between them is that TwitterRank introduces topic similarity between a user and his/her followers. IP-Influence [6] is a related algorithm similar to HITS [3], considering passivity of users, a measure of how different it is for other users to influence them. Another interesting approach leveraged probabilistic clustering and Gaussian-based ranking based on user features analysis [5].

TURank [10] measured the influence of users using ObjectRank [1] based on a user-tweet graph, which is most similar to our approach. The user-tweet graph used in this algorithm contains three relationships, following relationships between users, parent-child relationships between tweets, and posting-posted relationships between users and tweets. Different from these algorithms, we focus on topic-specific author ranking and our approach is based on a user-tweet graph according to retweeting relationships from readers to tweets and belonging relationships from tweets to authors, which emphasizes real interaction between users and tweets.

In our approach, the relationships between users are established through real tweets dynamically and indirectly, instead of directly using static following relationships, and some implicit user features are incorporated into our graph-based approach, which help improve the accuracy and effectiveness.

3 A User-Tweet Interaction Model

The user-tweet interaction model is the foundation of our author ranking approach. It is a weighted directed user-tweet graph $G = (V, E)$. V is a set of nodes which are of two kinds. Let $V_T = \{t_1, t_2, \dots, t_m\}$ be the set of *tweet nodes* representing tweets, and $V_U = \{u_1, u_2, \dots, u_n\}$ be the set of *user nodes* representing users. $V = V_T \cup V_U$. Here, tweet nodes only represent those original tweets, whose quality and popularity determine the influence degrees of their authors. $E = E^{U \rightarrow T} \cup E^{T \rightarrow U}$ is a set of edges of two type as well, where

$E^{U \rightarrow T} = \{e_1^{u \rightarrow t}, e_2^{u \rightarrow t}, \dots, e_p^{u \rightarrow t}\}$ is a set of edges from user nodes to tweet nodes, representing users respond to the tweets by retweeting with or without comments; and $E^{T \rightarrow U} = \{e_1^{t \rightarrow u}, e_2^{t \rightarrow u}, \dots, e_q^{t \rightarrow u}\}$ is a set of edges from tweet nodes to user nodes, representing the authorship of the tweets.

For an edge $e^{u \rightarrow t} \in E^{U \rightarrow T}$ from a user node to a tweet node, a weight is bounded according to whether the reader user retweets the tweet with or without comments.

$$Weight(e^{u \rightarrow t}) = \begin{cases} 1 & \text{if retweeting with comments} \\ \frac{1}{2} & \text{if retweeting without comments} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

It measures the contribution of a reader's behavior to the quality of a retweeted tweet. Retweeting with comments shows user's more interest in the tweet than without comments.

Also for an edge $e^{t \rightarrow u} \in E^{T \rightarrow U}$ from a tweet node to a user node (i.e., author node), the associated weight value signifies the contribution of the tweet to the influence degree of its author. Intuitively, more readers retweet the tweet, implying higher quality the tweet has, thus more influential the author is.

$$Weight(e^{t \rightarrow u}) = \frac{InDegree(t)}{C} \quad (2)$$

where $InDegree(t)$ is the in-degree of node t , representing tweet's retweeting number, and C is an adjustable positive integer, which could be set as the total number of user nodes in the user-tweet graph.

Figure 1 shows a user-tweet graph example.

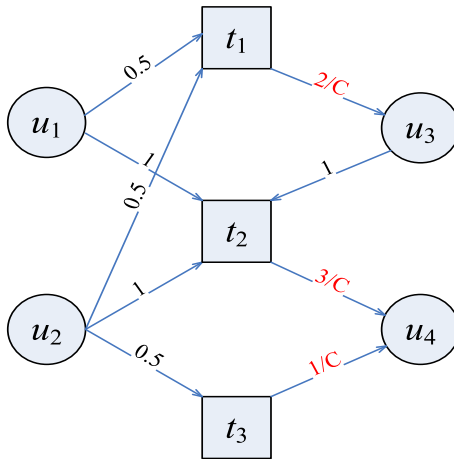


Fig. 1. A user-tweet graph example

4 Topic-Specific Author Ranking

Readers' interest behavior towards a tweet implies the quality of the tweet, which then determines the influence level of its author. To perform topic-specific influential authors' ranking, we first compute scores of tweet/user nodes in the directed weighted user-tweet graph. Here, the score of a tweet node reflects the quality of the tweet. The score of a user (as reader) node measures the topic-focus degree of the reader, and the score of a user (as author) node reflects the influence degree of the author. The computation process is iterative until all scores are converged.

4.1 Reader Score Reflecting Reader's Topic-Focus Degree

Consider there are two readers retweeting the same tweet. One of them only retweets tweets of the same topic, while the other retweets tweets of various topics. Obviously, on this topic, the voting of the former to the quality of the tweet has more referenced values than that of the later. We define the following reader score to measure the topic focus degrees of users as reader.

$$ReaderScore(u) = \begin{cases} \alpha \times \frac{RelatedOutDegree(u)}{OutDegree(u)} & \text{if } OutDegree(u) > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where $OutDegree(u)$ is the total number of tweets retweeted by user u , and $RelatedOutDegree(u)$ is the number of topic-related tweets retweeted by u . α is an adjustable basic reader score uniform for every user, which is set to the inverse of the number of users who post or retweet tweets on the topic in this study. The greater the value of $ReaderScore$ is, the more the reader user focuses on the topic.

4.2 Tweet Score Measuring Tweet's Quality

The score of a tweet is determined by the scores of users who respond to it. A random surfer model on the user-tweet graph is used to compute the score of tweet $t \in V_T$.

$$TweetScore_{n+1}(t) = p \times \frac{1}{|V_T|} + (1-p) \times \sum_{u \in V_U} (Weight(e^{u \rightarrow t}) \times \frac{UserScore_n(u)}{OutDegree(u)}) \quad (4)$$

Here, $TweetScore_{n+1}(t)$ is the score of tweet t computed in the $(n+1)$ -th iteration. p is the probability that a random surfer jumps to a random tweet instead of following the directed edges in the user-tweet graph, which is generally set to 0.15. $|V_T|$ is the number of topic-related tweets. $OutDegree(u)$ is the number of topic-related tweets retweeted by user u . $UserScore_n(u)$ is the score of the user computed in the n -th iteration.

4.3 User Score Containing Reader Score and Author Score

The user score during iteration consists of two parts, reader score and author score. We have defined the reader score of a user in Formula 3. Next we define the author score.

$$AuthorScore_{n+1}(u) = \sum_{t \in V_T} (Weight(e^{t \rightarrow u}) \times TweetScore_{n+1}(t)) \quad (5)$$

Here, $AuthorScore_{n+1}(u)$ is the author score of the user computed in the $(n+1)$ -th iteration. $TweetScore_{n+1}(t)$, the score of original tweet posted by this author, has been given in Formula 4. So the computation of $AuthorScore_{n+1}(u)$ corresponds to the observation that the influence of an author on the topic is determined by the quality of his/her original tweets.

Now the user score can be obtained through linear combination of reader score and author score as follows.

$$\begin{aligned} UserScore_{n+1}(u) &= ReaderScore(u) + AuthorScore_{n+1}(u) \\ &= \alpha \times \frac{RelatedOutDegree(u)}{OutDegree(u)} + \sum_{t \in V_T} (Weight(e^{t \rightarrow u}) \times TweetScore_{n+1}(t)) \end{aligned} \quad (6)$$

Algorithm 1. Topic-specific author ranking in micro-blog

Input: A topic-specific user-tweet graph $G = (V, E)$, where $V = V_T \cup V_U$

Output: A ranked list L of authors

```

1:  $p \leftarrow 0.15$ ; // the probability of a random surfer jumping in Formula 4
2:  $\alpha \leftarrow 1/|V_U|$ ; // the basic reader score in Formula 3
3: for each user  $u \in V_U$  do
4:    $ReaderScore(u) \leftarrow \alpha \times \frac{RelatedOutDegree(u)}{OutDegree(u)}$ ;
5:    $UserScore_0(u) \leftarrow 1.0$ ; // initial user score
6: end for
7:  $n \leftarrow 0$ ; // iteration indicator
8: repeat
9:   for each tweet  $t \in V_T$  do
10:     $TweetScore_{n+1}(t) \leftarrow p \times \frac{1}{|V_T|} + (1-p) \sum_{u \in V_U} (Weight(e^{u \rightarrow t}) \times \frac{UserScore_n(u)}{OutDegree(u)})$ ;
11:   end for
12:   for each user  $u \in V_U$  do
13:     $AuthorScore_{n+1}(u) \leftarrow \sum_{t \in V_T} (Weight(e^{t \rightarrow u}) \times TweetScore_{n+1}(t))$ ;
14:     $UserScore_{n+1}(u) \leftarrow ReaderScore(u) + AuthorScore_{n+1}(u)$ ;
15:   end for
16:    $n \leftarrow n + 1$ ;
17: until  $UserScore$  convergence
18:  $L \leftarrow \mathbf{Rank}(AuthorScore(u))$ ;
19: return  $L$ ;

```

4.4 Author Ranking

Note that $UserScore$ in Formula 6 is the score of user during iteration process, considering the contributions of users both as reader and as author. But when the iteration process is completed, $AuthorScore$ is used as the final ranking score of user, due to the observation that the influence of a user is determined by the quality of the original tweets posted by him/her. Algorithm 1 shows the core pseudocode of the approach.

We implement the algorithm using MapReduce framework to ensure the efficiency of large-scaled data processing. The first step is to construct a topic-specific user-tweet graph. Then the scores of tweets and users are computed iteratively until all scores are converged. At last, users are ranked by sorting their author scores.

Topic-related tweets can be simply extracted through keyword matching. Useful information related to these tweets are stored in a file for constructing the user-tweet graph, including authors, responders, behavior types of responders, and so on. In other words, the user-tweet graph exists in the form of a file. This file is placed on HDFS (Hadoop Distributed File System) as the input file of the following iterative processing.

5 Experiments

Our evaluation experiments were carried out on a hadoop cluster with 60 nodes, each with one 2.13GHz Intel Xeon Quad-Core processor, 32GB memory, and 12*146GB disks. The operation system is SUSE Linux Enterprise Server 10(x86_64). The hadoop version is 0.20.9.

Data set is from Tencent Micro-blog, which has more than 200 million users. We collected the information of all the tweets in the period of March 19th, 2011 to June 19th, 2011, a complete dataset of four months. The tweets are stored by date. The data file per day has approximately 4GB, and is placed on HDFS.

5.1 Empirical Evaluation of Author Ranking

We perform the topic-specific author ranking on three different topics - 1) NBA Finals, 2) Libya Crisis, and 3) DNF (Dos Not Finish, a popular online game) based on the data from June 1 to June 7. Table 1 shows each of the top 10 authors.

Topic 1 (“NBA Finals”). *NBA* is the official account of NBA in Tencent Micro-blog. *qq_nba* is the specific channel of Tencent focusing on NBA games. *titanduanxu*, *zhangweiping714*, *yugayujia*, and *yangyi* are all professional basketball analysts working for CCTV (China Central Television) or Sports Weekly. *linan10* is the assistant coach of the national basketball team, who was once a famous basketball player in China. The three remaining users, *yu_hua*, *bifeiyu*, and *sutongjs* went to USA and watched NBA final games live. They are all well-known professional writers with a large number of followers. Although their jobs

Table 1. Top 10 influential authors on three hot topics

Topic 1 (“NBA Finals”)	Topic 2 (“Libya Crisis”)	Topic 3 (“DNF”)
<i>NBA</i>	<i>qiuyongzhengABC</i>	<i>dnf</i>
<i>yu_hua</i>	<i>t_news</i>	<i>LyTouch</i>
<i>bifeiyu</i>	<i>laorong</i>	<i>hunangame</i>
<i>qq_nba</i>	<i>ktingstyle</i>	<i>jsgame</i>
<i>titanduanxu</i>	<i>qqnews</i>	<i>dnfcity</i>
<i>linan10</i>	<i>ludashi</i>	<i>dnfradio</i>
<i>zhangweiping714</i>	<i>C911001840</i>	<i>FoxySnow</i>
<i>yugayujia</i>	<i>video</i>	<i>qqgames</i>
<i>sutongjs</i>	<i>simapingbang</i>	<i>wb592319498</i>
<i>yangyi</i>	<i>zhanmin</i>	<i>shuang2</i>

are not about basketball, NBA games are their common hobby and their tweets about *NBA Finals* have high-quality.

Topic 2 (“Libya Crisis”). At the top of the list is *qiuyongzhengABC*, who is a war correspondent who stayed in Libya and posted many valuable tweets about Libya situation. *t_news*, *qqnews*, and *video* are all information media, broadcasting current news over the world. The six remaining users are active individual users who tweet about current affairs and express personal opinions. Among them, *laorong* and *ktingstyle* pay more attention to *Libya Crisis*; *C911001840* is interested in military affairs; and *simapingbang* and *zhanmin* are both media workers. Some opinions of them are sharp and strong, and motivate others to retweet and comment.

Topic 3 (“DNF”). *dnf*, *dnfcity*, and *dnfradio* are official accounts of the DNF game, focusing on its different aspects. *TG-hunan*, *TG-jiangsu*, and *qqgames* are game channels of Tencent. The four remaining users are active individual users related to the DNF game. *LyTouch* works in the game department of Tencent, and often releases some interesting information about *DNF*. *shuang2* is a popular DJ of DNF official broadcasting. *FoxySnow* and *wb592319498* are both senior players of DNF, and *FoxySnow* is a popular DJ of a game club radio as well.

From the results of these experiments, it can be seen that the effectiveness of our approach is satisfactory intuitively. In the next subsection, some results of rating and comparison are shown to quantify the effectiveness further.

5.2 Effectiveness of Author Ranking

Most existing graph-based algorithms for user ranking in micro-blog are variants of two classic algorithms, PageRank and HITS. In order to quantitatively examine the effectiveness of our approach, which is developed specifically for author ranking, we did comparison with the PageRank and HITS based approaches.

User’s PageRank. This approach is based on a user graph constructed according to the following relationships. Vertices refer to users, instead of pages in the original PageRank. Following relationships between users take the place of

hyperlink between pages. This approach has been used to compute the authority scores of users by [2].

User-tweet HITS. This approach is based on a bipartite graph whose vertices are composed of two independent sets, users and tweets. If a user retweets a tweet, there is an edge between the user node and the tweet node. During the iteration process, users are taken as hubs, while tweets are taken as authorities. When iteration is completed, total scores of original tweets are used as corresponding authors’ scores to measure the influence of users.

We conducted artificial rating over top 5 users obtained by the two reference algorithms and our approach on three different topics. The rating score of a user ranges from 1 to 5, which is determined by the relevance and quality of the contents generated by the user. 22 people, students, and employees of IT companies aging from 20 to 30 years old , took part in the rating. Each of them just rated the given users for their interested topics. We received 8 ratings for “NBA”, 11 for “Libya Crisis”, and 11 for “DNF”. Figure 2 shows the comparison results.

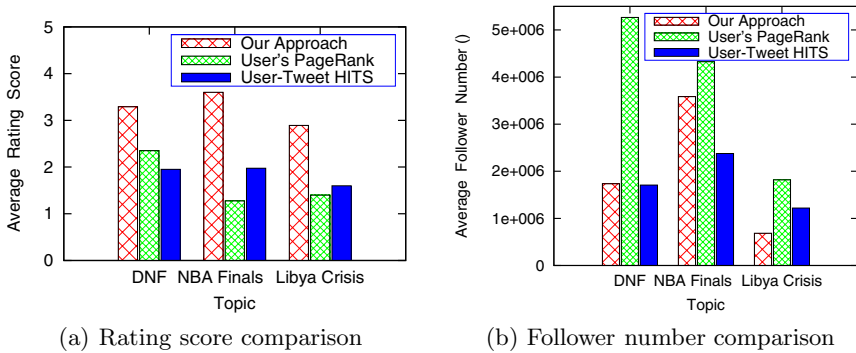


Fig. 2. Rating results comparison

As shown in Figure 2, for each topic, the average rating score of top 5 users from our approach is much higher than that from the other two methods. It is worth noting that the average follower number of top 5 users from User’s PageRank is the most for all the three topics, but the corresponding rating scores are the lowest for the other two topics out of three. In particular, for the topic “Libya Crisis”, although the average follower number of top 5 users from our approach is much fewer than that from the other two methods, the corresponding average rating score is the highest. This proves that the influence of a user on a specific topic is not necessarily proportional to his/her follower number.

5.3 Efficiency and Analysis of Timely Author Ranking

Sometimes, real-time author ranking is necessary, specially for a sudden event. It is meaningful to recognize the most relevant and influential authors as soon as

possible. For a long-standing topic, people may also want to find interesting authors at different stages. For example, Libya crisis suddenly got intensified when NATO carried out an air strike on March 20, 2011. Besides the authoritative news media, it is significant to find those individual users who originally tweet about the live situation in Libya. Timely author ranking solution on a daily or even hourly basis is very much desirable. Table 2 shows the daily ranking lists of top 5 authors on the “Libya Crisis” topic from March 19, 2011 to March 21, 2011 obtained by our approach.

Two authoritative news media, *t_news* and *qqnews*, appeared in the list of top 5 authors for all the three days. This is reasonable because people need to get news about “Libya Crisis” from authoritative news media. Particularly, on March 20 when NATO carried out an air strike in Libya, author *qiuyongzhengABC*, ranked in the second place, is a war correspondent who posted many valuable tweets about Libya situation; and on the following day, the war correspondent ascended to the first in the author list given by our algorithm. From the result of this experiment, it can be concluded that timely author ranking of our approach is effective.

Furthermore, we examine the stability of the ranking results on different topics. We conducted daily author ranking on the four topics (“NBA Finals”, “Libya Crisis”, “DNF”, and “iPhone”) from June 1, 2011 to June 7, 2011. For each top 10 authors from two adjacent days, the percentage of common authors (*pcu*) is used to measure the stability between the two days. Figure 3 shows the experimental results for seven consecutive days.

It shows that the greater the *pcu* is, the more stable the topic is. The most stable topic is “NBA Finals”, whose *pcu* is above 40%, even reaching 50%. “Libya” takes the second place, whose *pcu* ranges from 30% to 50%. The most unstable topic is “iPhone”, whose *pcu* is less than 10% throughout the seven days. “DNF” is different from the other three topics. Sometimes it is stable enough with a *pcu* reaching 50%; and sometimes it is quite unstable, whose *pcu* drops down to zero. These are due to the following reasons. “NBA Finals” is the most stable topic, because NBA games are a popular long-term sporting event, and some professionals or experts have been there for a long time, for example, basketball analysts. “Libya Crisis”, as an international event, has lasted for several months, which is similar to “NBA Finals”. So the authors of this topic is also stable, including news media and correspondents. The stability of “DNF” is inconsistent largely because business operation of game manufacturer leads to this uncertainty. As a popular electronic product, “iPhone” is quite a user-unstable

Table 2. Daily author ranking on the topic of “Libya Crisis”

March 19, 2011	March 20, 2011	March 21, 2011
<i>ktingstyle</i>	<i>qqnews</i>	<i>qiuyongzhengABC</i>
<i>t_news</i>	<i>qiuyongzhengABC</i>	<i>niudao</i>
<i>qqnews</i>	<i>t_news</i>	<i>gemagazine</i>
<i>christopherjing</i>	<i>ktingstyle</i>	<i>qqnews</i>
<i>mil - qq</i>	<i>huangdoc</i>	<i>t_news</i>

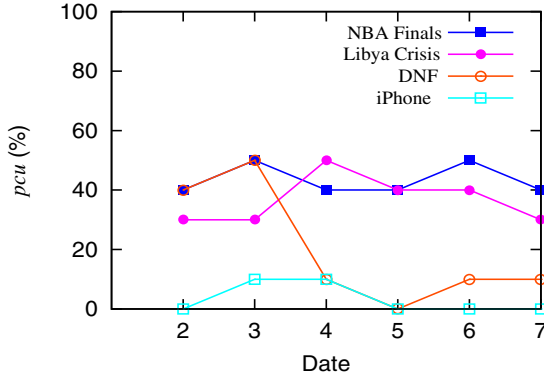


Fig. 3. Stability of daily author ranking results

topic because there is no official account or commercial operation of “iPhone” in Tencent Micro-blog.

5.4 Effect of Weight Settings in the User-Tweet Graph

We introduced $Weight(e^{u \rightarrow t})$ reflecting reader u ’s retweeting behavior to the quality of tweet t , and $Weight(e^{t \rightarrow u})$ reflecting tweet t ’s contribution to the influence of author u in the user-tweet graph. This aims to highlight the authors whose tweets are high-quality and popular, rather than those who post high-quantity but relatively low-quality tweets. We examine the effect of their settings on our topic-specific author ranking.

The left half of Table 3 shows the author ranking lists on the topic “NBA Finals” on June 1, 2011 under uniform and different weight settings from readers to tweets. Here, the uniform weight setting means that retweeting with/without comments is treated in the same way without distinction. That is, $Weight(e^{u \rightarrow t})$ is consistently set to 1 for retweeting behaviors. A different weight setting is 0.5 for retweeting without comments and 1 for retweeting with comments. With a uniform weight setting, *bifeiyu* ranks before *NBA*, and *yugayujia* ranks before *zhangweiping714*, which contrary to the rank result under different weight settings. *bifeiyu* is a famous writer, and NBA games are just his off-hour hobby; while *NBA* is the official account of National Basketball Association in Tencent

Table 3. Effect of $Weight(e^{u \rightarrow t})$ and $Weight(e^{t \rightarrow u})$

$Weight(e^{u \rightarrow t})$ on topic “NBA Finals”		$Weight(e^{t \rightarrow u})$ on topic “Libya Crisis”	
uniform	different	uniform	different
<i>bifeiyu</i>	<i>NBA</i>	<i>haowenhu</i>	<i>qqnews</i>
<i>NBA</i>	<i>bifeiyu</i>	<i>qqnews</i>	<i>qiuyongzhengABC</i>
<i>qq_nba</i>	<i>qq_nba</i>	<i>t_news</i>	<i>t_news</i>
<i>yugayujia</i>	<i>zhangweiping714</i>	<i>qiuyongzhengABC</i>	<i>ktingstyle</i>
<i>zhangweiping714</i>	<i>yugayujia</i>	<i>ktingstyle</i>	<i>huangdoc</i>

Micro-blog, focusing on various information about NBA games. Both *zhangweiping714* and *yugayujia* are well-known basketball analyst working for CCTV. But *zhangweiping714* provides live analysis for *NBA Finals* in USA, whose tweets are more attractive for NBA fans. Obviously, the later is more reasonable than the former. From this experiment, it can be concluded that distinguishing user behaviors is necessary and effective to recognize more authoritative users for a specific topic.

The right half of Table 3 shows the ranking lists of top 5 authors on the topic “Libya Crisis” on March 20, 2011 under uniform and different weights from tweets to authors. For the uniform weight setting, $Weight(e^{t \rightarrow u}) = 1$; while for the different weight setting, $Weight(e^{t \rightarrow u}) = \frac{InDegree(t)}{C}$ (C is set to the total number of user nodes in the user-tweet graph). *haowenhu* ranks first in the list obtained with a uniform setting. He posted a lot of low-quality tweets about Libya on March 20, 2011, and even plagiarized some tweets from *qiuyongzhengABC*, who is a war correspondent staying in Libya but only rank fourth in the list. In contrast, in the list obtained with different weight settings, the **spammer** *haowenhu* is kicked out from the top 5 users, and the real war correspondent *qiuyongzhengABC* ascends to the second place. Obviously, the latter is better than the former. So it is effective to recognize real influential authors, and exclude spammers through the edge weights from tweets to users.

6 Conclusion

In this paper, we propose a tweet-centric approach for topic-specific author ranking in micro-blog, which is based on a directed weighted user-tweet graph. The influence of an author is determined by the quality of tweets s/he has posted, which is in turn determined by the retweeting behaviors of their readers. We present a way to quantify the quality score of a tweet and influence score of an author. Implementation of the topic-specific author ranking algorithm is based on MapReduce framework, feasible to mass data processing and timely author ranking. Experimental result with the large-scaled real micro-blog from Tencent shows that our approach outperforms the PageRank and HITS based approaches in ranking accuracy.

We plan to combine this approach with event tracking and management in micro-blog, which could be convenient for reviewing historical events in the future. Applying the topic-specific author ranking to personalized tweet and author recommendation and tracking is also an interesting topic to be explored in the future work.

Acknowledgments. The work is funded by National Natural Science Foundation of China (60773156, 61073004), Chinese Major State Basic Research Development 973 Program (2011CB302203-2), and Tencent Research Fund.

References

1. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: Authority-based keyword search in databases. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, vol. 30, pp. 564–575. VLDB Endowment (2004)
2. Chen, C., Li, F., Ooi, B.C., Wu, S.: Ti: An efficient indexing mechanism for real-time search on tweets. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. ACM (2011)
3. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46(5), 604–632 (1999)
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Stanford Digital Library (1999)
5. Pal, A., Counts, S.: Identifying topical authorities in microblogs. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 45–54. ACM (2011)
6. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. In: Proceedings of the 20th International Conference on World Wide Web, pp. 113–114. ACM (2011)
7. Tunkelang, D.: A twitter analog to pagerank (2009), <http://thenoisychannel.com/2009/01/13/a-twitter-analog-to-pagerank/>
8. Welch, M.J., Schonfeld, U., He, D., Cho, J.: Topical semantics of twitter links. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 327–336. ACM (2011)
9. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twiterrank: finding topic-sensitive influential twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 261–270. ACM (2010)
10. Yamaguchi, Y., Takahashi, T., Amagasa, T., Kitagawa, H.: TURank: Twitter User Ranking Based on User-Tweet Graph Analysis. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 240–253. Springer, Heidelberg (2010)

An Algorithm for Sample and Data Dimensionality Reduction Using Fast Simulated Annealing

Szymon Lukasik¹ and Piotr Kulczycki²

¹ Department of Automatic Control and Information Technology,
Cracow University of Technology

ul. Warszawska 24, 31-155 Cracow, Poland

² Systems Research Institute, Polish Academy of Sciences,

ul. Newelska 6, 01-447 Warsaw, Poland

szymonl@pk.edu.pl,

kulczycki@ibspan.waw.pl

Abstract. This paper deals with dimensionality and sample length reduction applied to the tasks of exploratory data analysis. Proposed technique relies on distance preserving linear transformation of given dataset to the lower dimensionality feature space. Coefficients of feature transformation matrix are found using Fast Simulated Annealing - an algorithm inspired by physical annealing of solids. Furthermore the elimination or weighting of data elements which, as an effect of above mentioned transformation, were moved significantly from the rest of the dataset can be performed. Presented method was positively verified in routines of clustering, classification and outlier detection. It ensures proper efficiency of those procedures in compact feature space and with reduced data sample length at the same time.

Keywords: dimensionality reduction, sample reduction, linear transformation, fast simulated annealing, cluster analysis, classification, outlier detection.

1 Introduction

Modern data analysis has in its disposal a variety of methods based on both traditional and modern statistical techniques reinforced by soft computing procedures. Here, beside classical tools like fuzzy logic, neural networks and genetic algorithms, recent metaheuristics like particle swarm optimization, ant colony algorithms or bees optimization are frequently in use. Proper connection of algorithms' advantages enables their effective application in problems of contemporary knowledge engineering and data mining in particular. The subject of presented research is a concept of using nature-inspired Simulated Annealing algorithm [7] for the purpose of data dimensionality and sample size reduction.

Recently, the subject of data analysis are more and more frequently high dimensional datasets with huge sample lengths. It is a result of growing amount

of information stored in data warehouses. Extraction of knowledge from such datasets is a very complicated task. Difficulties include mainly limitations of computer systems' performance when considering huge samples and methodological obstacles of high dimensional data analysis. The latter is connected with properties of such datasets referred in bibliography as "curse of dimensionality" (this term was used for the first time by Bellman in the context of control systems design) [21]. It includes exponential grow of sample size needed to achieve proper efficiency of data analysis with increasing dimensionality, so called "empty space phenomenon" and vanishing of distances between close and distant points when using typical Minkowski norm.

To overcome above-mentioned problems adequate reduction procedures were developed. Sample length reduction is performed usually by means of sampling techniques [2] or advanced data condensation routines [14] and its expected result is mainly speeding up calculation time associated with data mining process. Dimensionality reduction can be performed in numerous ways. Let X to denote $n \times m$ data matrix:

$$X = [x_1 \ x_2 \ \dots \ x_m] \quad (1)$$

columns of which represent n dimensional sample elements for given probabilistic variable. Each dimension of such variable will be referred later in this paper as a feature. The aim of dimensionality reduction is a data transformation to a new $N \times m$ sized form, where N is significantly smaller than n . This can be achieved either by selecting most N significant features (feature selection) or by construction of a new set of N features based on the initial ones (i.e. by feature extraction). The second case is more general and will be considered in this work.

Among feature extraction procedures one can distinct: linear methods where synthesis of resulting dataset Y is performed by linear transformation:

$$Y = AX \quad (2)$$

with A being a transformation matrix of size $N \times n$ and nonlinear techniques where data transformation can be described by a nonlinear function $g : R^n \rightarrow R^N$ (or if such functional relationship does not exist). Details of feature transformation are usually established using some criterion which ensures maintaining critical data properties. It can be derived either from some general data characteristics (in unsupervised manner) or from the result of considered data analysis task (supervised feature extraction). One of the most widely used universal linear techniques of feature extraction is the principal components analysis (PCA). Conversely, multidimensional scaling (MDS) constitutes a typical representative of traditional nonlinear methods [6]. Studies on performance of routines belonging to both of above mentioned classes prove that even though nonlinear techniques possess more advanced mathematical background, they obtain often worse results in case of real-life datasets [12]. Apart from the performance the ability to create implicit mapping, which afterwards can be easily generalized to new data elements acquired dynamically, is also important in practical analytical tasks [10].

This paper introduces a new universal method of linear dimensionality reduction for use in exploratory data analysis. Dimensionality reduction is accomplished here by means of distance preserving linear transformation. The elements of the transformation matrix are to be determined using Fast Simulated Annealing. Additionally sample elements which as an effect of transformation significantly change their position could be eliminated or given lower weights. It can later serve in improvement of data analysis performance or sample length reduction.

The paper is organized as follows. Methodological preliminaries of the introduced method and its detailed description will be presented in the following Sections. As the performance of the technique under consideration was tested in clustering, classification and outlier detection procedures their short description will be given as well, followed by experimental results obtained in numerous testing trials. Finally some concluding remarks on the introduced method and planned further research will be given.

2 Methodological Preliminaries

2.1 Basic Exploratory Data Mining Tasks

First consider a problem of outlier detection. Such procedure is usually performed at the start of data exploration process to remove those elements from the sample which are found to be not representative. Usually it is performed by means of statistical approaches, e.g. using Grubbs test or Local Outlier Factor algorithm [3]. Measuring the performance of given procedure is difficult as usually it is not known in advance which element of the sample is atypical. However, if such knowledge is available, then the performance of the algorithm can be measured by:

$$I_{out} = \frac{c_o}{m} \quad (3)$$

where c_o is a number of correctly classified elements - either as an outlier or normal sample data point.

The task of cluster analysis is equivalent to such division of available data elements into subgroups (clusters) that elements belonging to each cluster are similar to each other and, at the same time, there is a significant dissimilarity between different clusters' elements. Numerous procedures have been developed to solve this problem. Among others K-means and DBSCAN algorithms can be named as popular ones [23]. If it is needed to compare different clustering solutions (or there exists a knowledge about cluster assignment) it is possible to use appropriate clustering indices e.g. Rand index:

$$I_{Rand} = \frac{a + b}{\binom{m}{2}} \quad (4)$$

with a and b being a number of data pairs which have been assigned to the same and different clusters in the both of analyzed solutions. If cluster number is

fixed, one can also try to form confusion matrix, align it properly to find cluster correspondence and calculate cluster preservation index I_{clust} [16].

Finally let us consider the task of classification, that is designating element $\tilde{x} \in R^n$ from the testing set to one of the fixed class with known set of representative patterns, similar to (II) (i.e. training set). Classification is often performed using instance-based learning methods e.g. k-nearest neighbor algorithm, along with more sophisticated statistical or computational intelligence procedures [19]. The efficiency of classification is evaluated by measuring its accuracy:

$$I_{class} = \frac{l}{m} 100\% \quad (5)$$

that is by a number of testing dataset elements l properly assigned to available classes, given as a ratio of overall dataset length. When precise division of the dataset into testing and training part is not explicitly given, one can use k-fold cross-validation, i.e. split available data into k sets and use one for evaluating purposes and the rest – for classifier learning. Whole process is usually repeated k times, although different variants of such validation can be found in the bibliography of the subject. Nevertheless in the case of cross-validation the average accuracy \bar{I}_{class} is usually reported as a final result.

2.2 Fast Simulated Annealing

Simulated Annealing (SA) is a heuristic algorithm which can be used in various optimization problems. Its idea is based on metallurgic annealing process. The SA algorithm incorporates iterative local search with individual acceptance criterion. By means of this criterion current algorithm's solution is established, typically with usage of solution quality index from two consecutive iterations and variable decreasing in time parameter called temperature of annealing. Moreover it is assumed that non-zero probability of worse solution acceptance should be enforced. This probability ought to decrease in time and enable the algorithm to escape from pitfalls of local minima. In most generic variants of the SA algorithm Metropolis rule is used as above mentioned criterion [7].

The algorithm in particular application demands specifying few functional elements like generation of initial and neighbor solutions, initial temperature and scheme of its changes, solution quality index and finishing criterion. Some general remarks concerning these issues were made in [15]. It is worth to mention as well that SA can be effectively used in continuous optimization with specific variants of the algorithm developed precisely for that purpose, e.g. Boltzmann Annealing, Fast Simulated Annealing and Adaptive Simulated Annealing [5].

Fast Simulated Annealing (in short: FSA), used in dimensionality reduction algorithm described here, is a strategy which employs random moves obtained by using multidimensional Cauchy distributed random numbers [18]. Global convergence of the algorithm to the optimal solution as iteration number t approaches infinity, is maintained by using conservative logarithmic annealing temperature schedule.

3 Algorithm Description

3.1 Dimensionality Reduction

Concept of the dimensionality reduction technique from n to predetermined N -dimensional space is based on linear transformation (2), given in detail by:

$$\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \vdots & \vdots & & \vdots \\ y_{N1} & y_{N2} & \dots & y_{Nm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{Nn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}. \tag{6}$$

Elements of transformation matrix A are found using Fast Simulated Annealing technique. Solution is represented as a vector:

$$z = [a_{11}, a_{12}, \dots, a_{1n}, a_{21}, a_{22}, \dots, a_{2n}, \dots, a_{N1}, a_{N2}, \dots, a_{Nn}]^T \in R^{nN}. \tag{7}$$

Solution quality index is given in the form of cost which is going to be minimized as a result of FSA algorithm. It can be represented in the form of:

$$g(z) = \sum_{i=1}^{m-1} \sum_{j=i+1}^m (d_{ij} - \delta_{ij}(z))^2 \tag{8}$$

or

$$g(z) = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^m d_{ij}} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{(d_{ij} - \delta_{ij}(z))^2}{d_{ij}} \tag{9}$$

where d_{ij} and δ_{ij} are distances (predominantly Euclidean) between sample points i and j in the initial and reduced feature space respectively. Both indices should enable achieving minimal difference of distances between sample elements in initial and reduced feature spaces, with additional emphasis put on small distances in the second cost function. Such formulations of solution quality indices are referred to as raw stress (8) and Sammon stress (9). Both were already used in nonlinear procedures of Multidimensional Scaling (4).

Initial solution is determined either randomly or by using feature selection algorithm presented in (13), with the second strategy being represented by two different variants. In general this alternative deterministic technique is based on the idea of feature space partition into clusters containing features which are similar to each other, with maximum information compression index being used as a similarity measure. Feature space clustering is performed using k -nearest neighbor algorithm, where k equal to $n - N$ should be assumed. As a result approximately N clusters are obtained. It is worth mentioning that a result of such initial solution's determination instead of being strictly fixed is customized to a real data structure. First approach of employing this feature selection algorithm to initial solution generation is based on N most representative features. The transformation matrix is formed in a way to retain them. It is achieved by using

1 and 0 properly as an indicative weights in the structure of A . Second strategy based on approach presented in [13] involves creating reduced feature set by linear combination of features included in each of “feature clusters”. To implement it, the solution of feature selection is stored in auxiliary vector $v \in R^n$. Each element of v characterizes the number of cluster to which corresponding feature from the initial feature space was assigned. This vector is then transformed into transformation matrix A using following rule: $a_{ij} = 1$ if $v_j = i$ and $a_{ij} = 0$ otherwise.

Initial temperature of FSA is determined from preliminary set of pilot runs and it ensures approximate 0.7 probability of worse solution acceptance in the introductory phase of the algorithm. Annealing ends after fixed number of iterations and as its result matrix A minimizing solution quality indices (8) or (9) and transformed dataset Y are obtained. In the case of classification task the reduction is performed for training dataset and reduced evaluation set is synthesized using transformation matrix formed as a result of such procedure.

3.2 Weighting and Sample Length Reduction

Linear transformation of feature space in the form presented in the previous Subsection can seriously affect some data elements’ relative position. Consequently the performance of data mining procedures in the reduced feature space can deteriorate significantly. As a countermeasure it is proposed to associate with each sample element a positive weight w_i normalized to ensure $\sum_{i=1}^m w_i = m$. Those weights are to be calculated using auxiliary parameters:

$$w_i^* = \frac{1}{\sum_{j=1, j \neq i}^m (d_{ij} - \delta_{ij})^2}, \quad (10)$$

and performing normalization:

$$w_i = \frac{mw_i^*}{\sum_{i=1}^m w_i^*} \quad (11)$$

for $i = 1, \dots, m$. Introduction of weights allows to take into account deformations in a relative data structure. Data elements with higher weights could then be treated as more adequate. Furthermore, one can use them as well to eliminate some data elements from the sample. It can be performed by removing instances with associated weights fulfilling following condition: $w_i < W$ where $W \in (0, +\infty)$ and then normalizing all weights ([11]). One can achieve in this way simultaneous dimensionality and sample length reduction with W serving as a data compression ratio.

4 Experimental Results

Proposed technique was verified for data exploration procedures based on four multidimensional example datasets taken from the UCI Machine Learning Repository [20].

Data dimensionality reduction routine was compared with PCA and unsupervised feature selection based on Evolutionary Algorithms [16] (first, best performing, variant of this algorithm was selected for this comparison). The latter was chosen for this study, because it employs Sammon stress as solution quality index. Heuristic procedure of Simulated Annealing as well as referenced techniques were executed in 10 independent trials (similarly to [16]). Each run was performed using 1000000 iterations as a stopping condition. Reduced feature space size N was selected according to [16], with the exception of *Vehicle* dataset for which standard PCA-based intrinsic dimensionality estimation was employed.

Table 1 summarizes results obtained for classification performed using five-fold cross validation and the nearest-neighbor classifier. Reported values include classification accuracy in the initial feature space, six variants of the FSA-based algorithm with different cost function and initial solution generation, as well as classification accuracy I_{class} obtained by referenced algorithms. It is important

Table 1. Dimensionality reduction for nearest-neighbor classification

	Glass		WBC	
	$m=214, n=9, N=4$		$m=683, n=9, N=4$	
	6 classes		2 classes	
	Accuracy [%]		Accuracy [%]	
	Average	Std. dev.	Average	Std. dev.
Initial FS	69.0	7.7	95.6	2.2
Raw, Linear combination	61.2	8.7	95.7	1.7
Raw, Feature selection	63.6	6.5	96.0	1.6
Raw, Random	62.1	7.7	96.0	1.4
Sammon, Linear combination	62.1	9.4	96.2	1.4
Sammon, Feature selection	64.5	4.4	95.9	1.5
Sammon, Random	61.4	9.9	96.0	1.2
EA-based [16]	64.8	4.4	95.1	0.8
PCA	57.6	9.9	96.3	1.8

	Wine		Vehicle	
	$m=178, n=13, N=5$		$m=846, n=18, N=5$	
	3 classes		4 classes	
	Accuracy [%]		Accuracy [%]	
	Average	Std. dev.	Average	Std. dev.
Initial FS	72.6	3.9	64.8	3.1
Raw, Linear combination	70.6	6.3	57.9	5.0
Raw, Feature selection	70.6	6.0	55.7	3.9
Raw, Random	76.0	5.1	66.4	3.9
Sammon, Linear combination	68.6	4.0	57.9	5.1
Sammon, Feature selection	70.9	6.0	56.2	6.4
Sammon, Random	75.4	6.1	64.9	3.5
EA-based [16]	72.8	1.0	60.8 [N=9]	1.5
PCA	70.9	8.4	46.9	5.5

to stress that for EA-based technique reduced feature set is synthesized using both training and testing sets. In the case of the algorithm being described here out-of-sample extension is used. It allows to transform testing set using transformation matrix synthesized for the training set. Nevertheless, results obtained are comparable to the ones achieved by referenced techniques. It can be noticed however, that it is difficult to select in advance which variant of the algorithm will reach highest-performance.

To test the possibility of sample size reduction classifier based on the kernel density estimators (KDE) was used [9], as its structure is very easy to modify to include weights [8]. In the considered case weighting scheme alone does not have a positive effect on classifier's performance. It can be used though to eliminate elements which as an effect of dimensionality reduction have a negative impact on data mining process. Elimination of elements with weights lower than 0.5 leads in some cases to the improvement of classification accuracy (see Table 2). It is predominantly observed when the sample size is too small to perform KDE-based classification reliably in the initial, high dimensional feature space. It confirms well-known fact that kernel density estimation is seriously affected by the curse of dimensionality [17].

Table 2. Dimensionality and sample size reduction for KDE-based classification

	Glass		WBC	
	Accuracy [%]		Accuracy [%]	
	Average	Std. dev.	Average	Std. dev.
Initial FS	60.5	7.6	95.0	2.0
PCA	52.6	8.9	93.0	2.8
Reduced	63.8	10.5	92.4	2.4
Reduced + Sample size reduction (Sample elements removed [%])	67.6 (8.7)	7.7 (1.8)	95.5 (10.3)	2.1 (2.1)

Finally cluster analysis and outlier detection experiments were performed. The preservation of cluster structure was indicated by cluster preservation index I_{clust} . In the case of outlier classification, its preservation was measured by (3). Values of both indices were reported for selected datasets in Table 3). Again, the technique under consideration achieved high accuracy of datasets structure preservation, comparable (or even better) to the one achieved by EA-based technique.

Table 3. Dimensionality reduction with cluster and outlier preservation

		Glass		WBC	
		Preserved [%]		Preserved [%]	
		Average	Std. dev.	Average	Std. dev.
Cluster preservation	Reduced	71.2	9.7	98.1	0.4
	EA-based [16]	69.3	4.9	94.7	2.3
Outlier Preservation	Reduced	95.4	0.9	88.1	1.1

5 Conclusion

This paper introduces new dimensionality and sample reduction technique designed for tasks of exploratory data mining. Introductory studies on method's performance prove that it offers promising solution quality in reference to the state-of-art principal components analysis procedure and similar heuristic based feature selection strategy. One should note however it is not specifically suited and designed for very high dimensional problems with huge sample sizes, as the optimization phase of FSA has significant computational complexity. It leads to exponential growth of computation time with increasing m . Nevertheless the method under consideration can be still used for data visualization and formulation of convenient data transformation, which can be later used in the data acquisition process. What is more, the possibility of practical implementation is significantly increased by employing simultaneous sample size reduction.

Further studies on the subject will concern various improvements in Fast Simulated Annealing scheme (e.g. statistic termination criterion of the algorithm). As Simulated Annealing can be effectively parallelized (refer to [1] and [11]) this area of research is going to be explored as well. It will allow the algorithm application for larger datasets. In addition prospective research will concern further improvements in sample size reduction scheme and its usage in various standard data mining algorithms.

References

1. Alba, E. (ed.): *Parallel Metaheuristics*. John Wiley & Sons, New York (2005)
2. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (1999)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. *ACM Computing Surveys* 41, 15:1–15:58 (2009)
4. Cox, T.F., Cox, M.A.A.: *Multidimensional Scaling*. Chapman & Hall, Boca Raton (2000)
5. Ingber, L.: Adaptive simulated annealing (ASA): Lessons learned. *Control and Cybernetics* 25(1), 33–54 (1996)
6. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1), 4–37 (2000)
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.: Optimization by Simulated Annealing. *Science* 220, 671–680 (1983)
8. Kowalski, P.A., Kulczycki, P.: Data Sample Reduction for Classification of Interval Information using Neural Network Sensitivity Analysis. In: Dicheva, D., Dochev, D. (eds.) *AIMSA 2010. LNCS*, vol. 6304, pp. 271–272. Springer, Heidelberg (2010)
9. Kulczycki, P.: Kernel Estimators in Industrial Applications. In: Prasad, B. (ed.) *Soft Computing Applications in Industry*, pp. 69–91. Springer, Heidelberg (2008)
10. Lee, J.L., Verleysen, M.: *Nonlinear Dimensionality Reduction*. Springer, Heidelberg (2007)
11. Lukasik, S.: Parallel Computing of Kernel Density Estimates with MPI. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) *ICCS 2007. LNCS*, vol. 4489, pp. 726–733. Springer, Heidelberg (2007)

12. van der Maaten, L.J.P.: Feature Extraction from Visual Data. PhD Thesis, Tilburg University, June 23 (2009)
13. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection using Feature Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(4), 301–312 (2002)
14. Pal, S.K., Mitra, P.: *Pattern Recognition Algorithms for Data Mining*. Chapman and Hall, London (2004)
15. Sait, S.M., Youssef, H.: *Iterative computer algorithms with applications in engineering*. IEEE Computer Society, Los Alamitos (1999)
16. Saxena, A., Pal, N.R., Vora, M.: Evolutionary methods for unsupervised feature selection using Sammons stress function. *Fuzzy Information and Engineering* 2(3), 229–247 (2010)
17. Silverman, B.W.: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London (1986)
18. Szu, H., Hartley, R.: Fast simulated annealing. *Physics Letters A* 122(3-4), 157–162 (1987)
19. Tan, P.-N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*. Pearson Addison-Wesley, Boston (2006)
20. UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/>
21. Verleysen, M., François, D.: The Curse of Dimensionality in Data Mining and Time Series Prediction. In: Cabestany, J., Prieto, A., Sandoval, F. (eds.) *IWANN 2005*. LNCS, vol. 3512, pp. 758–770. Springer, Heidelberg (2005)
22. Wand, M.P., Jones, M.C.: *Kernel Smoothing*. Chapman and Hall, London (1995)
23. Xu, R., Wunsch, D.C.: *Clustering*. Wiley, New York (2009)

An Investigation of Recursive Auto-associative Memory in Sentiment Detection

Saeed Danesh, Wei Liu*, Tim French, and Mark Reynolds

School of Computer Science and Software Engineering
The University of Western Australia
Australia
wei@csse.uwa.edu.au

Abstract. The rise of blogs, forums, social networks and review websites in recent years has provided very accessible and convenient platforms for people to express thoughts, views or attitudes about topics of interest. In order to collect and analyse opinionated content on the Internet, various sentiment detection techniques have been developed based on an integration of part-of-speech tagging, negation handling, lexicons and classifiers. A popular unsupervised approach, SO-LSA (Semantic Orientation from Latent Semantic Analysis), uses a term-document matrix to detect the semantic orientation of words according to their similarities to a predefined set of seed terms. This paper proposes a novel and subsymbolic approach in sentiment detection, with a level of accuracy comparable to the baseline, SO-LSA, using a special type of Artificial Neural Networks (ANN), an auto-encoder called Recursive Auto-Associative Memory (RAAM).

Keywords: Recursive Associative Memory, Artificial Neural Network in Natural Language Processing, Sentiment Detection, Semantic Clustering.

1 Introduction

The rise of blogs, forums, social networks and review websites in recent years has provided very accessible and convenient platforms for people to express opinion about topics of interest.

The techniques used to extract and explore such worthwhile information is often referred to as *Subjectivity and Sentiment Detection*. Such techniques in general rely on lexical, syntactic information, negation handling rules, and finally classifiers such as Support Vector Machines (SVM) [10]. Another popular spectrum in sentiment detection is the unsupervised approaches of Semantic Orientation through Association (SO-A) [15]. Assuming that the semantic orientation of a lexical unit is similar to that of its neighbours, SO-A is often based on statistical measuring of co-occurrence and contextual usages of words, such as Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA) [3].

* Corresponding author.

Because of its ability in dealing with high dimensionality of the term-document vector space, SO-LSA gains its popularity due to its simple, unsupervised nature and the ability of processing large volumes of structured and unstructured text. However it has some drawbacks. Firstly, it requires a large corpus for the results to be statistically valid. Secondly, its performance is highly dependent on the selection of the right seed terms. In addition, the meaning of a word captured through LSA, is the “average meaning” of all usages of the word in the corpus. Therefore, it cannot differentiate multiple meanings of a word based on its context. For example, the word pretty in “This is pretty bad” and “She is pretty” carry different meanings and perform different semantic roles. SO-LSA is not capable of dealing with such ambiguity, fundamentally due to the bags of words document model. The bags of words model, represents text as collection of words ignoring their order and positions in sentences. Subsequently, vector space models relies solely upon the lexical occurrences of words, thus having difficulty disambiguating words according to context within finer granularity. Associating words of similar syntactic or semantic roles is also challenging for such techniques.

The aim of the subsymbolic approach proposed in this paper is to achieve the same advantages of SO-LSA when compared to the classical approaches in the literature, yet trying to overcome some of its limitations as mentioned above and finally trying to model the ability of human brain in storing knowledge and reasoning based on experiences expressive of sentiment. The intuition behind this research is that not only words that have explicit positive or negative directions such as adjectives and adverbs can carry sentiment, but also some apparently factual words might carry sentiment as well. In other words, a word, no matter what part-of-speech it assumes in a sentence, brings contextual information that is specific to an individual. The training process of this research is to assert positive/negative experiences and then develop distributed representations of lexical units that carries hyper dimensional information, where sentiment is entrenched in. To achieve these goals, we use a distributed model called *Recursive Auto Associative Memory (RAAM)*, first introduced by Pollack [13] in 1990.

RAAM is a neural network and an *auto-encoder*, a system that can process input sentences represented as *trees* and operates at the tuple, noun phrase, verb phrase and sentence level. Given RAAM’s acknowledgment of sentence structure, ordering, roles of words and its ability of clustering based on syntactic and semantic similarities, in this paper, for the first time we investigate RAAM’s ability in sentiment detection. We designed two mechanisms, first using a single RAAM and then an ensemble of two RAAMs. The ensemble of two RAAMs is able to outperform SO-LSA with 70.7% accuracy versus 63.05% on the same dataset. In addition we report on the observed capability of RAAM in semantic clustering and polysemy disambiguation.

This paper is structured as follows. Section 2 discusses the state of the art. Section 3 introduces the structure of our system. Section 4 explains the data gathering process. Section 5 discusses neural networks role in semantic

clustering and how clustering is used for sentiment detection with some results and evaluations. Finally, the paper concludes in Section 6.

2 Related Work

2.1 Sentiment Detection

The most well known approaches in sentiment detection start with *opinion detection*, also known as *subjectivity classification*. Adjectives and adverbs are usually determined as good indicators of subjectivity. Once a subjective text is identified, a classifier can be used for sentiment detection [9,11,17]. In our research, we assume that the data to be processed are all subjective. That includes sentences with implicit or explicit sentiment.

Classification. Many of the tasks in sentiment detection are approached using classification techniques [10]. The main challenge in sentiment detection is finding the right features. Features can be based on presence of terms [17], n-grams [12], part-of-speech [16], syntax or negations [6].

Semantic Orientation. Another group of tasks in sentiment detection is based on identifying the semantic orientation of words. This can be achieved using a variety of techniques, for instance, generating *lexicons* or using *statistical information* about the words in documents. *SentiWordNet* [4] is an example of such specific lexicons made for sentiment detection. It is based on WordNet [5], a large English lexicon. Esuli and Sebastiani [4] associated entries in WordNet with scores of subjectivity, positivity and negativity. Turney and Littman [15] use the co-occurrence of words to infer the semantic orientation of them. For this purpose, they use PMI and LSA. PMI finds the co-occurrence of words by querying a search engine. LSA uses a term-document matrix. This matrix is then being processed by *Singular Value Decomposition* (SVD) in order to find statistical relations between words.

To employ LSA for sentiment detection, Turney and Littman [15], firstly select two sets of words, one positive and one negative.

- P, a set of words with positive semantic orientation, for example good, nice, excellent, positive, fortunate, correct, and superior.
- N, a set of words with negative semantic orientation for example bad, nasty, poor, negative, unfortunate, wrong, and inferior.

Secondly, they measure the strength of association between every word in the document to the positive and negative set using the equation below.

$$SO_{LSA}(w) = \sum_{p \in P} LSA(w, p) - \sum_{n \in N} LSA(w, n)$$

When $LSA(word_1, word_2)$ is positive, $word_1$ and $word_2$ are associated with each other. A higher LSA value represents closer association between the two words. When $LSA(word_1, word_2)$ is negative, one word always appear in the absence of the other (low association). If the value of $SO_{LSA}(w)$ is greater than zero, the word w has positive semantic orientation otherwise it has negative semantic orientation. Finally the average semantic orientation of all the words in the sentence (other than the stop words) are aggregated to determine the positivity or negativity of a sentence.

2.2 Artificial Neural Networks in Natural Language Processing

Artificial Neural Networks, also referred to as *Connectionism* or *Parallel Distributed Representation*, are models that have also been applied to natural language processing [2][7][8].

Recent work by Collobert and Weston [2], describes a single *convolutional* neural network capable of producing part-of-speech tags, noun phrase and verb phrase chunking, determining named entities, semantic roles and detecting semantically similar words from a given sentence through a unified architecture called *deep neural network*.

Deep neural network is based on the philosophy that natural language processing is a complicated task and if we want to perform a complex task, a complex system is required [2]. This is what we adopt in this research such that we move away from LSA and instead investigate the capabilities of more complex systems.

Using Artificial Neural Networks for the purpose of Natural Language Processing has a long history. The most related work to RAAM is *FGREP, Forming Global Representations with Extended back-Propagation* [7], dated back to 1988. FGREP is an auto-encoder and a model of subsymbolic processing that learns compressed representation of data through an encoding mechanism and uses dimensionality reduction to capture deeper correlations among entities, objects and their associated attributes.

FGREP, and in general, auto-encoders, can only process fixed sized representations. Therefore connectionist NLP systems are facing the question of how to feed a sentence with variable length to such an architecture. To address this shortcoming Collobert and Weston [2] proposed *word tag* and *sentence tag likelihood* and Pollack [13] proposed representing sentences as tree structures that could be transformed into *recursive* distributed representations of fixed size called *Recursive Auto Associative Memory*. RAAM is an auto-encoder that relies upon the compositionality nature of the natural language. Compositionality refers to the fact that language contains structural information that is hidden under the surface [18]. In particular, RAAM assumes that the input is a triplet of Action, Agent and Object.

3 System Design and Methodology

As illustrated in Figure 1, our system takes volumes of subjective text either positive or negative as inputs. Firstly, negation handling is carried out in the

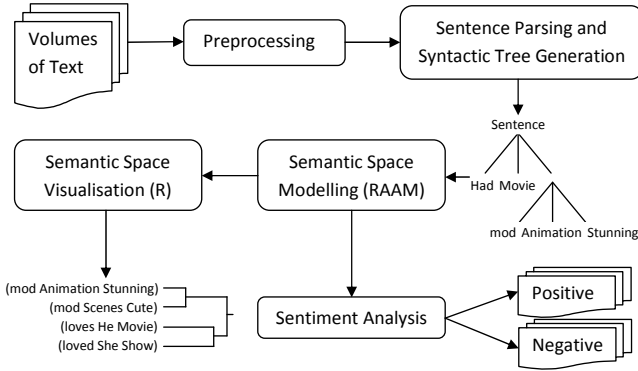


Fig. 1. System Design and Methodology

preprocessing task. Secondly we use a syntactic parser to produce the input data for the ensemble of two RAAMs. The parser begins by looking at the sentences based on their structure and syntax and determining where Subject (Agent), Predicate (Action), and Object in each chunk of text is located and how they all relate to one another.

Once the syntactic trees are generated, RAAM is used to learn the associations between the input triplets and generate a semantic space which could be visually observed by providing the outputs of the hidden layer to the R statistical package. The final step is sentiment analysis. For a given test sentence, our model simply compares the semantic orientation of a chunk of text to the positive and negative semantic spaces. We find this generalisation and similarity detection capability of RAAM very appropriate for sentiment detection. In fact, in general, this can be beneficial to any tasks that need to semantically splitting a corpus in two or multiple parts.

3.1 Implementation of RAAM

The overall structure of our RAAM network is a $kn-n-kn$ feed forward neural network with full connections between neighbour layers (Figure 2). Each word k has a fixed size representation (n bits) or n number of neurons. The number of words depends on the tree structure. In this research we used $k=3$ since the structure of trees are ACTION AGENT OBJECT triplets. We also used n bits for word representations based on the size of our lexicon. For instance to cover a dictionary with maximum size of 1024, we require at least 10 bits to represent a given word in binary.

Suppose that a tree T is defined as either a triplet of $\{P, L, M, R\}$ or ε (empty) where P is Parent; L, M and R are Left, Middle and Right children and each of them can be a tree T respectively. T is a leaf node if (left, middle and right) children of T are ε (empty). T is a root node if parent of T is ε (empty).

We used the encoding mechanism in the following algorithm, adopted from Wong [18], to transform a tree T into fixed size representations.

```

encode T;
if T has children
  encode L; encode M; encode R;
else
  look up for the code of T in the lexicon;
  feed L, M and R to the input and the desired output of RAAM;
  get values of the hidden layer and use as the representation of parent;
    
```

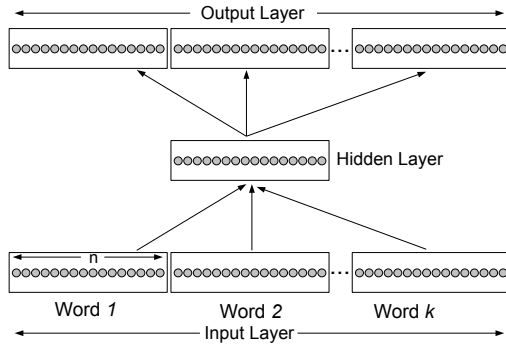


Fig. 2. RAAM model structure adopted from [11]

For instance, considering the sentence `The movie had stunning animation`, the input of RAAM will be a nested tree as shown in Figure 3. In order to convert `stunning animation` to a triplet, we use `mod` to represent a modifier.

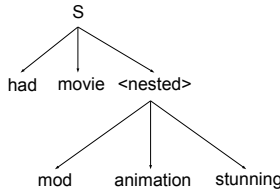


Fig. 3. Triplet tree for sentence: `The movie had stunning animation`

Once RAAM is trained with the triplet `mod animation stunning` as shown in Figure 4 (a), the value of the hidden layer is captured and used for training the entire sentence as described in Figure 4 (b).

3.2 A Single RAAM

We first performed the task of semantic clustering and sentiment classification using only one RAAM. Since there are many cases that positive and negative words in natural language appear within the same grammatical structure, for

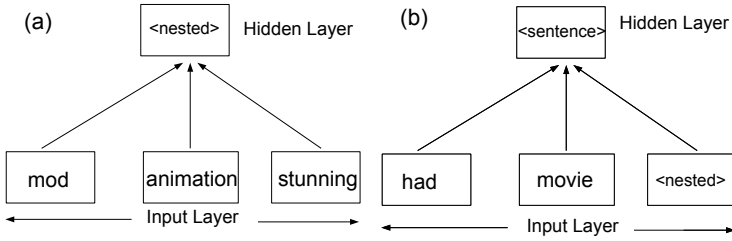


Fig. 4. RAAM Training, a two step process for sentence **The movie had stunning animation**

example **good** and **bad** in sentences **Movie is good** and **Movie is bad**, one RAAM fails to differentiate between positivity and negativity by putting positive and negative words in the same clusters. To overcome this shortcoming, we specified extra bit(s) of sentiment to indicate positivity (1.0) and negativity (-1.0) of each triplet in a sentence. As we will discuss more in section 5.2, it did not achieve a very high accuracy. This led us to propose an ensemble of two RAAMs.

3.3 A RAAM Ensemble

In order to use RAAM in sentiment detection, we propose an ensemble of two RAAMs as shown in Figure 5. One RAAM for testing and training a *positive corpus* and one other for testing and training a *negative corpus*. In order to measure the sentiment of a test chunk which can be a triplet, a noun phrase,

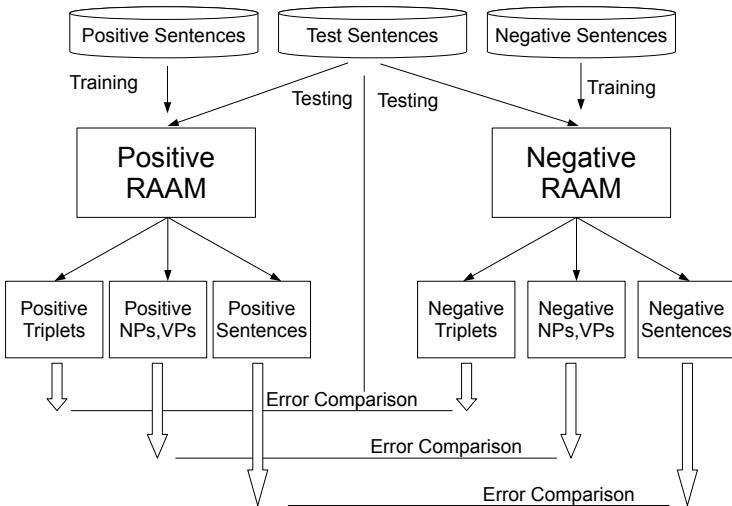


Fig. 5. Sentiment Detection using RAAM Model

a verb phrase or the entire sentence, we feed them into the both positive and negative RAAMs. For each input triplet, we compare the absolute sum of the errors generated from both RAAMs. Less error with positive RAAM comparing to the negative RAAM indicates the similarity of the test unit to the positive corpus. We use the same logic for the negative RAAM and the negative corpus. Less error with negative RAAM comparing to the positive RAAM indicates the similarity of the test unit to the negative corpus, and vice versa.

3.4 Triplet Tree Generation

Since the input of RAAM is a triplet tree, in order to convert an actual sentence like **The movie has stunning animation** to **(has movie (mod animation stunning))** automatically, we use the Stanford parser¹ and a triplet extraction algorithm adopted from Rusu et al. [14]. The *predicate*, *subject*, *object* and their *attributes* are extracted from a sentence to make up a triplet compatible to RAAMs required input structure which is **ACTION AGENT OBJECT**.

For handling negations, we use simple bi-gram features. The word *not* or any other *negation modifier* plus *the next word* will be considered as a new word, for example **not-like** or **seldomly-interesting**. For instance the sentence **I do not like Audrina** will be preprocessed as **I do not-like Audrina**.

4 Data Collection and Parameter Determination

Sentences input to RAAM are represented as triplets. As a result, the sentiment being detected is at the triplet level, including simple triplets, noun and verb phrases and sentences.

The benchmark data available from previous experiments for example by Pang and Lee² are mostly at the document and snippe³ level, there is also the need of co-reference resolution because the subjects are often left out which results in ill-formed sentences. Movie review being positive for instance does not mean that every sentence in that document is positive. To avoid co-reference resolution of the ambiguity brought in by long documents, we choose not to use Pang & Lee's benchmark movie review data instead we resort to use movie reviews from *Twitter*. *Twitter* is a micro-blogging social networking website that has a large, up to date, rich and rapidly growing data bank in the form of text messages. The advantage of using *Twitter* reviews is that a *tweet* is 140 characters at the most which suits our sentiment detection approach at the sentence level. In order to extract data from twitter, we used Twitter search engines like *tweetfeel*⁴ to gather opinionated and labeled movie reviews.

We evaluate our system based on a corpus made of 2000 sentences, with half positive and half negative sentences having implicit or explicit sentiments.

¹ <http://nlp.stanford.edu/software/lex-parser.shtml>

² <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

³ a paragraph, a few lines of text.

⁴ <http://www.tweetfeel.com>

To use the binary representation of words as the inputs of a neural network, we convert zeros to -1.0 and ones to 1.0 (double) for the network with sigmoid function that works in the interval of [-1.0, 1.0]. Therefore a word like **beautiful** with binary representation (1 0 1 0 1 0) will be converted to (1.0 -1.0 1.0 -1.0 1.0 -1.0) as the input and the desired output pattern. We stop the training for each RAAM when the absolute sum of the errors reached the number of triplets multiplied by error size of 0.01.

5 Experiments and Results

5.1 Similarity Detection

The cluster dendrograms shown in Figure 6(a) to (d) demonstrate some of the capabilities of RAAM in capturing a combination of lexical, syntactic and semantic similarities of triplets. In order to produce these dendrograms, we used the R statistical package. The data used for cluster analysis is the output of the hidden layer of RAAM for each triplet after the training phase was completed.

Figure 6(a) shows RAAM’s ability in clustering sentences that are structurally similar. The left word **was** is the same but the network clusters **HarryPotter** and **cast** together slightly separated from **action** and **ending** as attributes of a movie. Figure 6(b) demonstrates the ability of our system to cluster based on the similarity of verbs **created** and **did** and similar concept and structure of the entire sentences and also the nested triplets. In Figure 6(c), the entire cluster has the general meaning **I thought the movie was good**. The sentence with ID 1612 has 3 triplets while the others have two. The nested phrase **the movie is good** is the same with the nested triplets of the other sentences. In Figure 6(d), the clustering is based on the similarity of the structures and nested triplets while the **movie** and **actress** concepts are in separate clusters.

5.2 Sentiment Detection

The ability of RAAM in grouping similar entities and structures as described in section before, could help when more additional data is not provided for

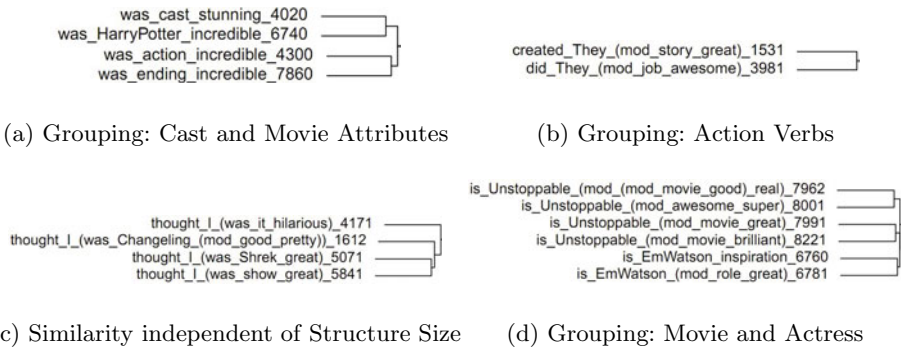


Fig. 6. Sentence Structure Similarity and Semantic Similarity

training. Assuming the sentences **The actress wears cute glasses**, and **The actors wear nice hats**, it can predict the sentiment of a sentence like **Mary and John wear nice glasses** as positive. In other words, it is able to represent knowledge beyond the training set [13].

To estimate the performance of RAAM in sentiment detection, we used 10 folds cross validation. Data was divided into 10 folds with each folds containing randomly selected sentences from the dataset. Figure 7(a), illustrates the accuracy, precision, recall and true negative rate of a corpus with 2000 sentences, half positive and half negative. It describes only a single RAAM results in sentiment detection. The average accuracy for the 10 folds is only 59.25%. Figure 7(b) is the results of our proposed RAAM ensemble. We compare our achievement against SO-LSA as the baseline. We used the Java implementation of LSA from the *S-Space* package [5]. To implement SO-LSA, we wrote an interface to calculate the semantic orientation of each word in a corpus with the defined positive and negative seed terms from LSA generated semantic space based on cosine similarity. Firstly we used two sets of words, ignoring their frequency:

P={good, funny, awesome, cute, love}

N={bad, sad, awful, stupid, hate}

Secondly we only selected the top five positive and negative words based on their frequency considering the same range of frequency for the pairs of positive and negative.

P={awesome, amazing, pretty, cute, fun}

N={boring, terrible, horrible, lame, awful}

We achieved the best accuracy for SO-LSA based on the second experiment. As for the dimension of LSA (the size of the semantic space), we found out that the best performance is achieved by dimension sizes of 10 (75%) and 7 (75.5%) on the entire corpus. The dimension size in LSA is also comparable to the representation size of 10 of our RAAM.

We measured the accuracy of RAAM in sentiment detection based on the aggregation of the errors of all the triplets in a sentence generated from positive and negative RAAM or only the triplets that are one complete sentence, ignoring their nested triplets. Aggregating the errors of all triplets of a sentences increases the true positives and true negatives. So we use this measure in calculating precision, recall, accuracy and true negative rate in Figure 7.

In Figure 7(b), on average we achieved 70.7% accuracy for the ensemble of two RAAMs. In this figure, we have also the average results of SO-LSA for 10 folds (when LSA is blind to 10% of the data like RAAM). In this experiment, RAAM appears to be 7% more accurate.

⁵ <http://code.google.com/p/airhead-research/>

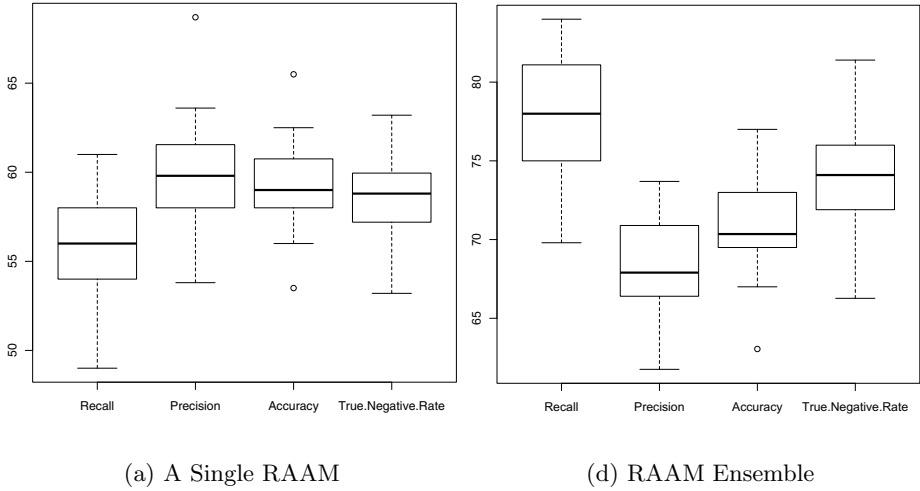


Fig. 7. 10-Fold Results for Recall, Precision, Accuracy and True Negative Rate

5.3 Results on Ambiguous Words

In our movie review corpus, sentences may carry both implicit and explicit sentiments. Words can be ambiguous, for example, the word **pretty** is used as intensifiers in Table 1, while in general, it can also have the positive meaning of **nice looking**. For instance, Table 1 demonstrates how RAAM and SO-LSA classified some of the sentences into positive and negative.

Comparing to the basic SO-LSA, RAAM is able to capture structures with words that do not have explicit sentiment but as a group of words, together they have positive and negative meanings. For example the sentence **I must watch the show** has no sentiment according to SO-LSA (the verb **watch** is neutral).

One other capability of RAAM is classifying based on roles and entities. After training with **The movie made me cry** as positive and **She made me cry** as negative, the testing sentences **Shrek made me cry** was detected as positive however **You make me cry** was detected as negative because **Shrek** is a movie and **You** is closer to **She** as a pronoun.

RAAM also makes generalisations on the sentiments associated with objects and entities just like an individual’s mind. In our training data, positive reviews about **Shrek** for example, were twice as much as negative ones. Our analysis showed that RAAM knows **Shrek** as a positive entity.

Table 1. RAAM versus SO-LSA in Sentiment Detection

Sentence	RAAM	SO-LSA
It was pretty good	+	+
It was pretty boring	-	+
It was pretty horrible	-	+

6 Conclusion and Future Work

In this research we designed mechanisms of using RAAM for sentiment detection. As the first attempt on subsymbolic sentiment detection we achieved an accuracy comparable to the popular SO-LSA. With subsymbolic natural language processing we can cluster the part of the speech that share similarities, both syntactically and semantically. The results are still preliminary and the current ensemble is only capable of binary classification. Currently, we are working to extend this work to analyse the ability of RAAM in multi-class classification, in particular, to classify stock forum postings into buy, strong buy, hold, sell, strong sell, etc.

Acknowledgement. The authors are grateful to Australia China Council on the partial financial support from the Australia-China Grant of ACC00003 (2011-2012).

References

1. Chalmers, D.: Syntactic transformations on distributed representations. *Connection Science* 2(1), 53–62 (1990)
2. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th International Conference on Machine Learning (ICML)*, Helsinki, Fabianinkatu, pp. 160–167 (July 2008)
3. Dumais, S., Landauer, T.: A solution to platos problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review* 104, 211–240 (1997)
4. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: *Proceedings of Language Resources and Evaluation (LREC)*, Genoa, Italy (May 2006)
5. Fellbaum, C., et al.: *WordNet: An electronic lexical database*. MIT press, Cambridge (1998)
6. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22(2), 110–125 (2006)
7. Miiikkulainen, R., Dyer, M.: Forming global representations with extended back-propagation. In: *IEEE International Conference on Neural Networks*, pp. 285–292. IEEE (1988)
8. Moisl, H.: Artificial neural networks and natural language processing. In: *Encyclopedia of Library and Information Science*, pp. 148–162 (2003)
9. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, pp. 115–124 (June 2005)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
11. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 271. Association for Computational Linguistics, Barcelona (2004)

12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, PA, USA (July 2002)
13. Pollack, J.: Recursive distributed representations. *Artificial Intelligence* 46(1-2), 77–105 (1990)
14. Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., Mladenić, D.: Triplet extraction from sentences. In: 10th International Multiconference Information Society-IS, Ljubljana, Slovenia, pp. 8–12 (July 2007)
15. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)* 21, 315–346 (2003)
16. Whitelaw, C., Argamon, S., Garg, N.: Using appraisal taxonomies for sentiment analysis. In: Proceedings of the First Computational Systemic Functional Grammar Conference. University of Sydney, Sydney (2005)
17. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning subjective language. *Computational Linguistics* 30(3), 277–308 (2004)
18. Wong, C.: Recursive auto-associative memory as connectionist language processing model: training improvements via hybrid neural-genetic schemata. Master's thesis, City University of Hong Kong (2004)

APPECT: An Approximate Backbone-Based Clustering Algorithm for Tags

Yu Zong^{1,2}, Guandong Xu³, Ping Jin^{1,*}, Yanchun Zhang³,
EnHong Chen², and Rong Pan⁴

¹Department of Information and Engineering, West Anhui University, Luan, 237012, China

²Department of Computer Science and Technology, University of Science and Technology, Hefei, 230036, China

³Center for Applied Informatics, Victoria University, PO Box 14428, VIC 8001, Australia

⁴Department of Computer Science, Aalborg University, Denmark

Nick.zongy@gmail.com, jinping@wxc.edu.cn

Abstract. In social annotation systems, users label digital resources by using tags which are freely chosen textual descriptions. Tags are used to index, annotate and retrieve resource as an additional metadata of resource. Poor retrieval performance remains a major problem of most social tagging systems resulting from the severe difficulty of ambiguity, redundancy and less semantic nature of tags. Clustering method is a useful tool to address the aforementioned difficulties. Most of the researches on tag clustering are directly using traditional clustering algorithms such as K-means or Hierarchical Agglomerative Clustering on tagging data, which possess the inherent drawbacks, such as the sensitivity of initialization. In this paper, we instead make use of the approximate backbone of tag clustering results to find out better tag clusters. In particular, we propose an APProximate backboneE-based Clustering algorithm for Tags (APPECT). The main steps of APPECT are: (1) we execute the K-means algorithm on a tag similarity matrix for M times and collect a set of tag clustering results $Z=(C^1, C^2, \dots, C^m)$; (2) we form the approximate backbone of Z by executing a greedy search; (3) we fix the approximate backbone as the initial tag clustering result and then assign the rest tags into the corresponding clusters based on the similarity. Experimental results on three real world datasets namely MedWorm, MovieLens and Dmoz demonstrate the effectiveness and the superiority of the proposed method against the traditional approaches.

Keywords: Approximate backbone, Tag clustering, Social annotation systems.

1 Introduction

With the development of Web2.0 application services, tag-based services, e.g., Del.icio.us¹, Last.fm², and Flickr³, have undergone tremendous growth in the past

* Corresponding author.

¹ <http://del.icio.us>

² <http://www.last.fm>

³ <http://flickr.com>

years. Tags are simple, uncontrolled and ad-hoc labels that are assigned by users to describe or annotate any kind of resource [1]. The low technical barrier of social tagging systems and the easy use of tagging have attracted a large amount of research interest. The user-contributed tags are not only an effective way to facilitate personal organization but also provide a possibility for users to search for needed information.

Recently tagging has been widely used in social annotation systems for many applications [2-4]. The common usage of tags in these systems is to add the tagging attribute as an additional feature to re-model users or resources over the tag vector space, and in turn, making tag-expanded collaborative filtering recommendation or personalized recommendation. However, as the tags are of syntactic nature, in a free style and do not reflect sufficient semantics, the problems of redundancy, ambiguity and less semantics of tags are often incurred in all kinds of social tagging systems. For example, for one resource, different users will use their own words to describe their feeling of likeness, such as “favorite, preference, like” or even the plural form of “favorites”; and another obstacle is that not all users are willing to annotate the tags, resulting in the severe problem of sparseness. In order to deal with these difficulties, recently clustering methods have been introduced into social tagging systems to find meaningful information conveyed by tag aggregates. The aim of tag clustering is to reveal the coherence of tags from the perspective of how resources are annotated and how users annotate in the tagging behaviors. Undoubtedly, the tag cluster form is able to deliver user tagging interest or resource topic information in a more concise and semantic way, which, in some extent, to handle the problems of tag sparseness and redundancy, in turn, facilitating the tag-based recommender systems. Thus this demand mainly motivates the research of tag clustering in social annotation systems. In general, the tag clustering algorithm could be described as to: (1) define a similarity measure of tags and construct a tag similarity matrix; (2) execute a traditional clustering algorithm such as K-means [5], or Hierarchical Agglomerative Clustering [6] on this similarity matrix to generate the clustering results; (3) abstract the meaningful information from each cluster and do recommendation. Although the experimental results have shown the success of these algorithms, the inherent drawbacks of the traditional clustering algorithm, such as the sensitivity of initialization and high computational cost etc, are the main reason, which affects the performance of the proposed tag clustering algorithms.

In order to alleviate the inherent drawbacks of tag clustering algorithms, in this paper, we propose an APProximate backbonE-based Clustering algorithm for Tags (APPECT). The motivation of our method is based on the phenomenon that the common part of several clustering results, derived by executing a traditional clustering algorithm for several times, captures the most significant entities within a clustering process from the perspective of optimization. Using this common part, particularly coined *Approximate Backbone* is able to improve the clustering results [7]. In the context of tag clustering, we especially extract the approximate backbone, i.e., the core tags to form the initial tag clusters and conduct the tag clustering.

The main steps of APPECT are: we firstly define a similarity measure for tags by considering the resource and user aspects simultaneously and construct a similarity matrix SM ; secondly, we execute a traditional clustering algorithm on SM for M times

to obtain a collection of clustering results $Z = \{C^1, C^2, \dots, C^M\}$; thirdly, we design an algorithm, FIND_AB based on greedy search method to find out the approximate backbone from Z ; eventually, we fix the approximate backbone as the initial clustering results, and assign the rest tags to the most appropriate clusters based on the similarity. In order to evaluate the effectiveness of the proposed approach, we conduct experiments on three real world tagging datasets. The contributions of our work are as follows:

- we define a similarity measure for tags by considering the resource and users aspects simultaneously;
- we introduce the concept of approximate backbone for tags that can capture the core tags to form various tag clusters from the multiple execution of traditional clustering, and devise a greedy search algorithm to form the approximate backbone;
- we propose a new approximate backbone based clustering algorithm for tags, in which we make use of the approximate backbone of tags;
- we conduct comparative experiments on three real world datasets to evaluate the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows. We review the related work in Section 2 and introduce the preliminaries in Section 3. The details of APPECT algorithm are discussed in Section 4. Experimental evaluation results are reported in Section 5. Section 6 concludes this paper and outlines the future work.

2 Related Work

In the past years, many studies have been carried out on tagging clustering. [5, 6] demonstrated how tag clusters serving as coherent topics can aid in the social recommendation of search and navigation. Astrain et al. firstly combines a syntactic similarity measure based in a fuzzy automaton with ϵ -moves and a cosine relatedness measure, and then design a clustering algorithm for tags to find out the short length tags [8]. In general, tags lack organizational structure limiting their utility for navigation. Simpson proposes a hierarchical divisive clustering algorithm to release these influence of the inherent drawback of tag data [9]. In [10], an approach that monitors users' activity in a tagging system and dynamically quantifies associations among tags is presented and the associations are then used to create tags clusters. Zhou et al. propose a novel method to compute the similarity between tag sets and use it as the distance measure to cluster web documents into groups [11].

In [12] topic relevant partitions are created by clustering resources rather than tags. By clustering resources, it improves recommendations by distinguishing between alternative meanings of query. While in [13], clusters of resources are shown to improve recommendation by categorizing the resources into topic domains. A framework named Semantic Tag Clustering Search, which is able to cope with the syntactic and semantic tag variations, is proposed in [14]. P. Lehwark et al. use Emergent-Self-Organizing Maps (ESOM) and U-Map techniques to visualize and cluster tagged data

and discover emergent structures in collections of music [15]. State-of-the-art methods suffice for simple search, but they often fail to handle more complicated or noisy Web page structures due to the key limitations. Miao et al. propose a new method for record extraction that captures a list of objects in a more robust way based on a holistic analysis of a Web page [16]. In [17], a co-clustering approach is employed, that exploits joint groups of related tags and social data resources, in which both social and semantic aspects of tags are considered simultaneously. The common characteristic of aforementioned tagging clustering algorithm is that they use a traditional clustering algorithm on tag dataset to find out the similar tag groups. In [18], however, the authors introduce FolksEngine, a parametric search engine for folksonomies allowing specifying any tagging clustering algorithm. In the similar way, Jiang et al, make use of the concept of ensemble clustering to find out a consensus tag clustering results of a given topic and propose tag groups with better quality [19].

The efficient way which improves tag clustering result is to use the common parts of several tag clustering results. *Approximate Backbone*, the intersection of different solutions of a dataset, is often used to investigate the characteristic of a dataset [20-22]. Zong et al. use approximate backbone to deal with the initialization problem of heuristic clustering algorithm [7]. In this paper, we firstly define approximate backbone to capture the common tags among the tag clustering results derived by executing a traditional clustering algorithm several times on a tag similarity matrix. And then fix approximate backbone as the initial tag clustering result, and assign the rest tags into the one with highest similarity.

3 Preliminaries

3.1 Social Tagging System Model

In this paper, our work is to deal with the tagging data. A typical social tagging system has three types of objects, users, tags and resources which are interrelated with one another. Social tagging data can be viewed as a set of triples [23]. Each triple (u, r, t) represents a user u annotates tag t to resource r . A social tagging system can be described as a four-tuple, where exists a set of users, U ; a set of tags, T ; a set of resources, R ; and a set of annotations, A^N . We denote the data in the social tagging system as D and define it as $D = \langle U, R, T, A^N \rangle$. The annotations are represented as a set of triples contains a user, tag and resource defined as: $A^N \subseteq \langle u, r, t \rangle : u \in U, r \in R, t \in T$. Therefore a social tagging system can be viewed as a tripartite hyper-graph [24] with users, tags and resources represented as nodes and the annotations represented as hyper-edges connecting users, resources and tags.

3.2 Similarity Measure

In the framework of tag clustering, the tag similarity plays an important role. In real applications, in order to compute the similarity, we usually start from the tripartite graph of social annotations to compose a two-dimensional, e.g., the resource-tag

matrix by accumulating the frequency of each tag along users. In this expression, each tag is described by a set of resources, to which this tag has been assigned, i.e., $t_i = (wr_{i1}, wr_{i2}, \dots, wr_{im})$, where wr_{ik} denotes the weight on resource dimension r_k of tag t_i , which could be determined by the occurrence frequency. In this manner, the similarity between any two tags on resource is defined as Definition 1.

Definition 1. Given two tags $t_i = (wr_{i1}, wr_{i2}, \dots, wr_{im})$ and $t_j = (wr_{j1}, wr_{j2}, \dots, wr_{jm})$, the similarity is defined as the Cosine function of angle between two vectors t_i and t_j :

$$Simr(t_i, t_j) = \frac{t_i \cdot t_j}{\|t_i\| \cdot \|t_j\|} \quad (1)$$

Definition 1 only considers the relationship between tags and resources, however the relation between users and tags was neglected. Tags are freely used by users to reflect their opinions or interests on various resources. If we simultaneously consider the similarity of tags from both resource and user perspective, a complete relationship between tags will be captured.

To realize this, we build another user-tag matrix from the tripartite graph by using the similar approach, i.e., $t_i = (wu_{i1}, wu_{i2}, \dots, wu_{im})$, where wu_{ik} denotes the weight on user dimension u_k of tag t_i , which could be determined by the occurrence frequency. Definition 2 gives the similarity of tags on users.

Definition 2. Given two tags $t_i = (wu_{i1}, wu_{i2}, \dots, wu_{im})$ and $t_j = (wu_{j1}, wu_{j2}, \dots, wu_{jm})$, the similarity is defined as the Cosine function of angle between two vectors t_i and t_j :

$$Simu(t_i, t_j) = \frac{t_i \cdot t_j}{\|t_i\| \cdot \|t_j\|} \quad (2)$$

Furthermore, the similarity of tags is defined as the line combination of $Simr$ and $Simu$, as defined in Definition 3.

Definition 3. Given two tags t_i and t_j , $Simr(t_i, t_j)$, and $Simu(t_i, t_j)$, the overall similarity between t_i and t_j is defined as:

$$Sim(t_i, t_j) = \alpha \cdot Simr(t_i, t_j) + (1 - \alpha) \cdot Simu(t_i, t_j) \quad (3)$$

where $0 \leq \alpha \leq 1$ is a tuning factor for $Sim(t_i, t_j)$. If $\alpha = 0$, the similarity only accounts the importance from the user aspect, otherwise, the contribution of resource aspect dominates the similarity, i.e. when $\alpha = 1$. For whole tags, we calculate the similarity between each tag pair according to Definition 1-3, and a similarity matrix SM_{tag} is constructed.

3.3 Approximate Backbone of Tag Clustering Results

Most tag clustering algorithms utilize traditional clustering algorithms, such as K-means, on the tag similarity matrix to conduct the tag clustering result C . As known, clustering is an unsupervised learning process. One execution of the traditional clustering algorithm on tag set could be regard as one knowledge extraction process, and the clustering result is the learning result of tag data. If we use the common information of several clustering results derived by running a traditional clustering algorithm on tag dataset to initialize the clustering algorithm, the more robust clustering result will be obtained. In this section, we introduce the concept of approximate backbone to capture the common information from several tag clustering results. Assume that a traditional clustering algorithm is run for M times on tag similarity matrix and a tag clustering result collection $Z = \{C^1, C^2, \dots, C^M\}$ is generated, where, $C^m, m=1, \dots, M$ denotes the tag clustering result of the m -th clustering, the common part of Z is defined as the approximate backbone of tag clustering results.

Definition 4. Given a collection of tag clustering results $Z = \{C^1, C^2, \dots, C^M\}$, where $C^m = \{C_1^m, C_2^m, \dots, C_K^m\}$ denotes the m -th clustering result, and $C_k^m, k=1, \dots, K$ denotes the k -th cluster in the m -th clustering result. The approximate backbone of Z is defined as $AP(Z) = \{ap_1, ap_2, \dots, ap_K\}$, where $ap_k, k=1, \dots, K$ satisfies the following conditions:

- (1) $|ap_k| \geq 2$ and (2) all elements in $ap_k, k=1, \dots, K$ must be co-occurred among Z for M times.

According to Definition 4, the common part of Z is captured.

4 The Details of APPECT

In this section, we first propose an algorithm with greedy search for capturing the approximate backbone from the tag clustering collection Z , and then we discuss the details of APPECT, at last, we analyses the time cost of APPECT.

4.1 Capture the Approximate Backbone

In order to capture the approximate backbone of tag clustering result collection Z , we propose an algorithm named Find_AB (Find Approximate Backbone by using Set intersection operation). We assume that each cluster $C_k^m \subset Z$ is consisted by tag id, e.g., $C_k^m = \{\#2, \#5, \dots, \#338\}$. In the tag clustering result collection, there are C_M^K possible combinations of clusters. Each cluster C_k^m must be intersected with the other clusters in $C^{m'}, m'=1, \dots, M, m \neq m'$, so the whole traversal search method is very time consuming. In this section, we thus design the algorithm Find_AB via greedy search. Algorithm 1 shows the main steps of Find_AB.

Algorithm 1. Find_AB

Input: tag clustering result collection Z

Output: approximate backbone of Z , $AP(Z) = \{ap_1, ap_2, \dots, ap_k\}$

- (1) $AP(Z) \leftarrow \emptyset$;
 - (2) Randomly select a baseline C^m from Z ;
 - (3) for $k = 1, \dots, K$
 - (3.1) intersect C_k^m with $C_{k'}^{m'}$ in C^m of Z , where $m' = 1, \dots, M, m \neq m'$ and $k' = 1, \dots, K$;
 - (3.2) if $|C_k^m \cap C_{k'}^{m'}| \geq 2$ { $ap_k = C_k^m \cap C_{k'}^{m'}$; $AP(Z) = AP(Z) \cup ap_k$; }
 - (4) return $AP(Z)$;
-

Algorithm Find_AB shown in algorithm 1 gives the main steps of finding the approximate backbone set $AP(Z)$. The first step is to randomly select a baseline partition from Z . Each cluster of the baseline partition will be intersected with all the clusters in the rest partitions, as shown in step 3, and the data objects co-occurring in the same cluster are find out. The intersection of data objects in each cluster will be regard as the $AP(Z)$ and then returned.

4.2 Details of APPECT

After the approximate backbone of tag clustering result collection Z is generated, we fix them as the initial clustering result, that is, $C_{ini} \leftarrow AP(Z)$. We define the tags in cluster $C_k \subset C_{ini}$, $k = 1, \dots, K$ as the core tags, because they are co-occurred in M times clustering. And eventually, we assign the rest tags into the corresponding clusters based on the highest similarity to the cluster centers.

The main steps of APPECT are shown in algorithm 2.

Algorithm 2. APPECT

Input: the tag similarity matrix, SM , executing times, M ;

Output: the tag clustering result, C ;

- (1) generate tag clustering result collection, Z , by executing a traditional clustering algorithm for M times;
 - (2) find the approximate backbone $AP(Z)$ from Z by running Find_AB;
 - (3) fix approximate backbone of Z as the initial clustering result, that is, $C_{ini} \leftarrow AP(Z)$;
 - (4) Assign the rest tags into the corresponding clusters in C_{ini} based on the highest similarity;
 - (5) return $C \leftarrow C_{ini}$;
-

Algorithm 2 consists of three main steps: firstly, we generate the tag clustering result collection Z by running a traditional clustering algorithm, such as, K-means, on the similarity matrix SM . The main aim of this step is to prepare sufficient tag clustering result for finding the substantially co-occurred tags. Secondly, according to Definition 4, we find out the co-occurred tags which are conveyed by the approximate backbone $AP(Z)$. In this step, we use a greedy search method to capture the co-occurred tags by running the Find_AB algorithm. At last, we first fix the approximate backbone $AP(Z)$ as the initial clusters, and then assign the rest tags into the corresponding clusters with the highest similarity.

4.3 Time Analysis of APPECT

The first step of APPECT executes a traditional clustering algorithm for M times to generate the tag clustering result collection Z . Assume that the running time of a traditional clustering algorithm is $O(\cdot)$, it needs $M \times O(\cdot)$ time cost for the M times running. In the second step of APPECT, the approximate backbone of Z is derived by executing Find_AB and its time cost is $O(MK)$. At last, we assign the rest tags into corresponding clusters with the highest similarity and the time cost at most is $O(NK)$.

According to the above discussion, the total time cost of APPECT is $M \times O(\cdot) + O(MK) + O(NK)$, where M is the number of running time, N is the number of tags, K is the number of clusters and $K \ll N, M \ll N$. The last two parts of the time cost have the linear relationship with M , N and K , so the main time cost of APPECT depends on the first part, i.e., $M \times O(\cdot)$. In real experiments, we execute the K-means algorithm to generate Z . In this situation, the time cost of APPECT is $M \times O(tNK) + O(MK) + O(NK)$, where $t \ll N$ is the iterative number of K-means.

5 Experiments and Evaluations

To evaluate our approach, we conducted experiments on three real world datasets: MedWorm⁴, MovieLens⁵ and Dmoz⁶. We performed the experiments using an Intel Core 2 Duo CPU (2.4GHz) workstation with 4G memory, running Windows XP. All algorithms were implemented in Matlab 7.0.

5.1 Data Sets

The first dataset is extracted from the crawled MedWorm article repository during April 2010. After stemming out the entity attributes from the data, four data files, namely user, resource, tags and quads, are obtained as the source datasets. Here we only use the fourth data, which presents the social annotations where each row

⁴ <http://www.medworm.com>

⁵ <http://www.movielens.org>

⁶ <http://www.michael-noll.com/dmoz100k06/>

denotes a user u annotates resource r by using tag t . The second dataset is MovieLens which is provided by GroupLens. This dataset consists of three files - movies.dat, rating.dat and tags.dat. The tags.dat has the same format as the quads in MedWorm dataset, which we utilize to conduct the experiments. The last dataset dmoz100k06 is a large research dataset about document metadata based on a random sample of 100,000 web documents from the Open Directory combined with data retrieved from Delicious.com/ Yahoo!, Google, and ICRA. For our study, we use the tagging data available at del.icio.us, one of the most popular social bookmarking services. The data from del.icio.us was collected over a period of three weeks in December 2006. This dataset contains 13,771 documents with 25,311 tags by 5,016 users. The statistical results of these three datasets are listed in Table 1. These three datasets are pre-processed to filter out some noisy and extremely sparse data subjects to increase the data quality. Lin [25] has indicated the fact that the distributions of tags, resources and users are subject to power distribution. This observation is justified by the statistical results of these three datasets shown in Table 1.

Table 1. Statistics of Experimental Datasets

Property	MedWorm	MovieLens	Dmoz
Number of users	949	4,009	5,016
Number of resources	261,501	7,601	13,771
Number of tags	13,507	16,529	25,311
Total entries	1,571,080	95,580	97,587
Average tags per user	132	11	123
Average tags per resource	5	9	11

5.2 Evaluation Metrics

The aim of tag clustering is to assign the tags serving for the similar function into the same cluster. Here we assume that a better tag cluster is composed of similar tags, which are dissimilar to tags belonging to other different tag clusters. In particular, we use Similarity and Dissimilarity to validate our method.

Definition 5. Given the tag clustering result $C = \{C_1, C_2, \dots, C_K\}$, its similarity is defined as

$$SimiM(C) = \frac{1}{K} \sum_{k=1}^K \frac{2 \cdot Sim(t_i, t_j)}{|C_k| \cdot (|C_k| - 1)}, \quad t_i, t_j \in C_k \tag{4}$$

Definition 6. Given the tag clustering result $C = \{C_1, C_2, \dots, C_K\}$, its dissimilarity is defined as

$$DissimiM(C) = \frac{1}{K} \sum_{k=1}^K \frac{Dism(k)}{|C_k| \cdot (|T| - |C_k|)} \tag{5}$$

where $Dism(k) = \sum_{k'=1}^K Sim(t_i, t_j)$, $t_i \in C_k, t_j \in C_{k'}, k \neq k'$, and $|T|$ is the total number of tags.

According to the requirement of tag clustering, it is obvious that the higher Similarity value and smaller Dissimilarity value indicate better tag clustering results.

5.3 Experimental Results and Discussions

The first step of APPECT is to generate a tag clustering result collection by running a traditional clustering algorithm on the similarity matrix for M times. The number of M will have a direct impact on the quality of tag clustering results. In order to reveal this relationship, we conduct experiments on three datasets shown in Table 1 and the experimental results are shown in Fig. 1(a), (b). From the $SimiM(C)$ plot in Fig. 1(a), we can find that the $SimiM(C)$ plots of three datasets significantly change when M increases from 2 to 5 with step 1, while these plots gradually stabilize with M varying from 6 to 10. Likewise, in Fig. 1(b), the changes of $DisimiM(C)$ of these three datasets exhibit the similar varying trends with $SimiM(C)$, that is, three curves are dramatically descending with M increasing from 2 to 5, and they change slightly after $M > 5$. The change trends of $SimiM(C)$ and $DisimiM(C)$ have shown that the quality of tag clustering result does not heavily rely on the running times of K-means algorithm after an appropriate execution number is reached. In the other words, when the approximate backbone of Z has captured enough common information of tag clusters, the capability of approximate backbone on improving the tag clustering becomes diminished. According to the above discussion, we should select a proper M value which not only guarantees the quality of tag clustering result, but also, optimizes the time cost of APPECT. In the rest experiment, we set M as 5 based on the results of Fig. 1.

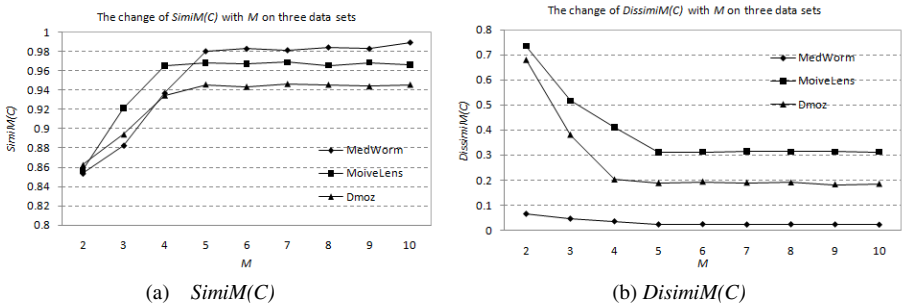


Fig. 1. The $SimiM(C)$ and $DisimiM(C)$ changes with the increase of M

In order to illustrate the relationship between α and the clustering results, i.e., $SimiM(C)$ and $DisimiM(C)$, we conduct experiments on three real world data sets under α changing from 0 to 1 by step 0.1. The clustering results are shown in Fig. 2.

From Fig. 2(a), we can find that the $SimiM(C)$ values of three datasets are reaching higher when α changing between [0.4, 0.6] than in other ranges. And the same phenomenon on $DisimiM(C)$ is observed in Fig. 2(b). This observation implies that the user and resource feature contribute almost evenly to the similarity measure, i.e., α varying in the range of [0.4, 0.6]. According to the experimental result, in this paper,

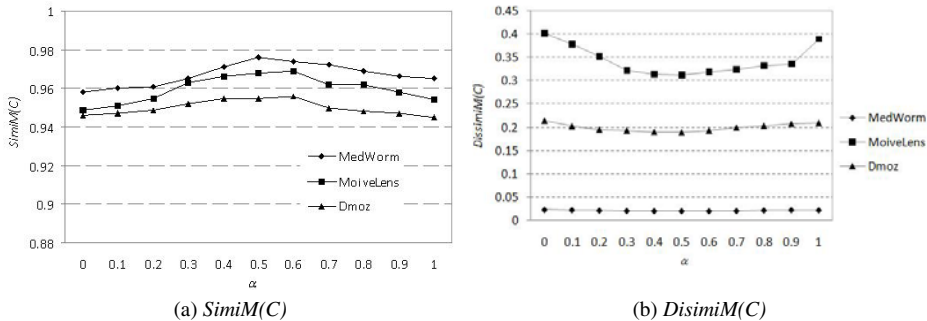


Fig. 2. The $SimiM(C)$ and $DissimiM(C)$ changes with the tuning parameter of α

we set $\alpha = 0.5$. Our proposed tag clustering algorithm aims to capture the tag groups with higher $SimiM(C)$ value and lower $DissimiM(C)$ value. We conduct the comparative studies of K-means, Single-link [26] and APPECT on three real world datasets in terms of $SimiM(C)$ and $DissimiM(C)$. Since we use a combined similarity measure which takes the resource aspect and user aspect to define the similarity of tags, we also carry out comparisons of three clustering algorithms on different similarity measure matrices, e.g., K-means(Simr) denotes that we run K-means algorithm on the similarity matrix derived by Definition 1, and K-means(Sim) indicates that we execute K-means on the similarity matrix generated by using Definition 3, and so on. For each tag dataset, we first execute the K-means algorithm on two different similarity measures for $M=5$ times to obtain the tag clustering result collection, Z ; and then, we calculate the $SimiM(C)$ and $DissimiM(C)$ according to Define 4 and 5, respectively and the tag clustering result with the highest $SimiM(C)$ and the lowest $DissimiM(C)$ value in Z is saved as the tag clustering result of K-means; at last, we form the approximate backbone of Z , and use it to find the tag clustering result of APPECT. For single-link clustering, we execute it on two different similarity matrix of three tag dataset to generate the corresponding K clusters and then calculate the clustering result quality in terms of $SimiM(C)$ and $DissimiM(C)$. The experimental results are shown in Fig. 3.

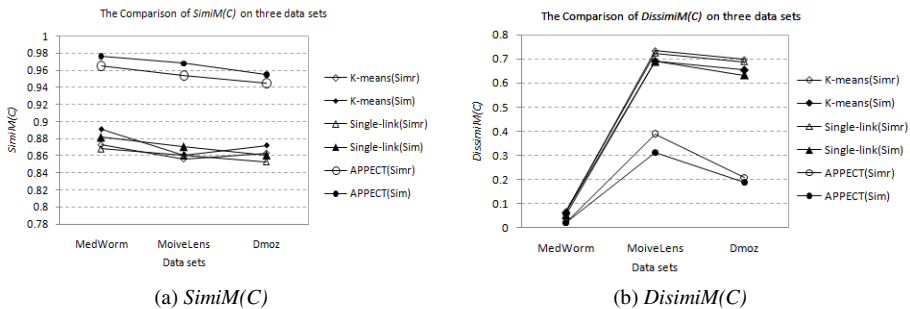


Fig. 3. The comparison of $SimiM(C)$ and $DissimiM(C)$ of three clustering algorithms with different tag similarity measures on three datasets

For each clustering algorithm, we see that the quality of tag clustering results using Simr is consistently lower than those of Sim, i.e., the $SimiM(C)$ values using Simr are smaller than those using Sim for each compared clustering algorithm, while the $DissimiM(C)$ values are larger than those of Sim. The reason to this is because that the combined similarity measure is able to better capture the overall similarity for the tag clustering. Furthermore, the tag clustering result of APPECT on the three datasets are better than those of K-means and Single-link. As such we can conclude that the combined similarity measure benefits the similarity computation and the common core tags captured by the approximate backbone have a significant effort on improving the clustering quality of tags.

Fig. 4(a) and (b) give the improvement comparisons of $SimiM(C)$ and $DissimiM(C)$, respectively. We can find from Fig. 4(a) that the $SimiM(C)$ of APPECT has increased by at least 9.7% in comparison to those of K-means on three datasets. In particular, the improvement on MoiveLens dataset is most significant. Similarly, the $SimiM(C)$ of APPECT has an almost 10% increase against those of Single-link. Fig. 4(b) shows the improvements of $DissimiM(C)$ of APPECT –K-means and APPECT–Single-link. The results shown in Fig. 4 reveal that the total quality of APPECT has a nearly 35% overall (considering both the improvement of $SimiM(C)$ and $DissimiM(C)$) improvement than those of K-means and Single-link on three datasets.

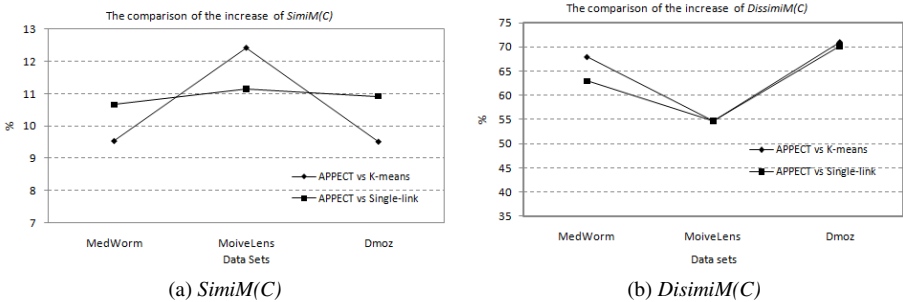


Fig. 4. The improvement comparisons of $SimiM$ and $DissimiM$ between APPECT and K-means, Single-link on three datasets

In order to evaluate the efficiency of three compared clustering algorithms, we compared the CPU time of them, and the results are shown in Fig. 5. According to the CPU time lines in Fig. 5, the K-means algorithm has the lowest time cost, which depends on the tag set size. For our algorithm, we first need generate Z by running K-means for M times and then find out the approximate backbone. So the time cost of APPECT is nearly M times than K-means. There have two ways to deal with the time cost of APPECT: (1) If we have the tag clustering result Z archived in offline stage, then APPECT could run more faster; (2) using the parallel computation method is another option for reducing the time cost of APPECT. The time cost of single-link algorithm is $O(N^2)$, so it is the most computational expensive one.

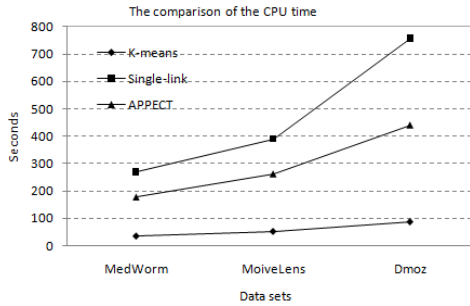


Fig. 5. Comparisons of CPU time of three clustering algorithm on three datasets

6 Conclusion and Future Work

Under social tagging systems, a typical Web2.0 application, users label digital data sources by using tags which are freely chosen textual descriptions. Tags are used to index, annotate and retrieve resource as an additional metadata of resource. The poor retrieval performance remains a major problem of most social tagging systems resulting from the severe difficulty of ambiguity, redundancy and less semantic nature of tags. Clustering is a useful tool to deal with these difficulties. Inspired by the common information of tag clustering results collection Z derived by running a traditional clustering algorithm for several times has the potential in improving the quality clustering results, we propose a new clustering algorithm for tags, named APPECT. We first define the approximate backbone to describe the common information of Z ; and then, devise an algorithm, named Find_AB based on the greedy search to capture the approximate backbone from Z to form the initial clusters; at last, the complete tag clusters are built up by assigning the rest tags into the corresponding initial clusters, based on the highest similarity. Experimental results on three real world datasets have shown that APPECT has the superiority in improving the quality of tag clustering. Future work could focus on reducing the time of cost of the proposed algorithm.

Acknowledgements. This work has been supported by the Nature Science Foundation of china under the Grant No 60775037 and 60933013. the Ph.D. Programs Foundation of Ministry of Education of China under Grant No. 20093402110017. the Nature Science Foundation of Anhui Education Department under Grant No. KJ2009A54, KJ2011Z321.

References

- [1] Zong, Y., Xu, G.D., Jin, P., et al.: A local information passing clustering algorithm for tagging systems. In: The Second Workshop on Social Networks and Social Media Mining on the Web, Hong Kong, pp. 333–343 (2011)

- [2] Duraio, F., Dolog, P.: Extending a hybrid tag-based recommender system with personalization. In: SAC 2010: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1723–1727. ACM, New York (2010)
- [3] Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag Recommendations in Folksonomies. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 506–514. Springer, Heidelberg (2007)
- [4] Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: SAC 2008: Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 1995–1999. ACM, New York (2008)
- [5] Gemmell, J., Shepitsen, A., Mobasher, M., Burke, R.: Personalization in folksonomies based on tag clustering. In: Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (July 2008)
- [6] Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized recommendation in social tagging systems using hierarchical clustering. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 259–266. ACM (2008)
- [7] Zong, Y., Jiang, H., Li, M.C.: Approximate backbone guided reduction clustering algorithm. *Journal of Electronics and Information Technology* 31(2), 2953–2957 (2009)
- [8] Astrain, J.J., Echarte, F., Córdoba, A., Villadangos, J.: A Tag Clustering Method to Deal with Syntactic Variations on Collaborative Social Networks. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 434–441. Springer, Heidelberg (2009)
- [9] Simpson, E.: Clustering tags in enterprise and web folksonomies. HP Labs Technical Reports, citeulike: 2545406 (2008)
- [10] Boratto, L., Carta, S., Vargiu, E.: RATC: A Robust Automated Tag Clustering Technique. In: Di Noia, T., Buccafurri, F. (eds.) EC-Web 2009. LNCS, vol. 5692, pp. 324–335. Springer, Heidelberg (2009)
- [11] Zhou, J.L., Nie, X.J., Qin, L.J., et al.: Web clustering based on tag set similarity. *Journal of Computers* 6(1), 59–66 (2011)
- [12] Matteo, N.R., Peroni, S., Tamburini, F., et al.: A parametric architecture for tags clustering in folksonomic search engines. In 9th international Conference on Intelligent Systems Design and Applications, Pisa, Italy, pp. 279–282 (2009)
- [13] Chen, H., Dumais, S.: Bringing order to the web: Automatically categorizing search results. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 145–152. ACM (2000)
- [14] van Dam, J., Vandic, D., Hogenboom, F., Frasinca, F.: Searching and browsing tagspaces using the semantic tag clustering search framework. In: 2010 IEEE Fourth International Conference on Semantic Computing (ICSC), pp. 436–439. IEEE (2010)
- [15] Lehwarck, P., Risi, S., Ultsch, A.: Visualization and clustering of tagged music data. *Data Analysis, Machine Learning and Applications*, 673–680 (2008)
- [16] Miao, G., Tatemura, J., Hsiung, W., Sawires, A., Moser, L.: Extracting data records from the web using tag path clustering. In: Proceedings of the 18th International Conference on World Wide Web, pp. 981–990. ACM (2009)
- [17] Giannakidou, E., Koutsonikola, V., Vakali, A., Kompatsiaris, Y.: Co-clustering tags and social data sources. In: The Ninth International Conference on Web-Age Information Management, pp. 317–324. IEEE (2008)
- [18] Nicola, R.D., Silvio, P., Fabio, T., et al.: Of mice and terms: Clustering algorithms on ambiguous terms in folksonomies. In: Proceeding of the 2010 ACM Symposium on Applied Computing SAC 2010, pp. 844–848 (2010)

- [19] Jiang, Y.X., Tang, C.J., Xu, K.K., et al.: Core-tag clustering for web2.0 based on multi-similarity measurements. In: The Joint International Conference on Asia-Pacific Web Conference (APWeb) and Web-Age Information Management (WAIM), Suzhou, China, pp. 222–233 (2009)
- [20] Zou, P., Zhou, Z.H., Chen, G.L.: Approximate backbone guided fast ant algorithm to QAP. *Journal of Software* 16(10), 1691–1698 (2005)
- [21] Jiang, H., Zhang, X.C., Chen, G.L.: Exclusive overall optimal solution of graph bipartition problem and backbone compute complexity. *Chinese Science Bulletin* 52(17), 2077–2081 (2007)
- [22] Jiang, H., Zhang, X.C., Chen, G.L.: Backbone analysis and algorithm design of QAP. *Chinese Science* 38(01), 1–14 (2008)
- [23] Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., He, X.: Document recommendation in social tagging services. In: Proceedings of the 19th International Conference on World Wide Web, pp. 391–400. ACM (2010)
- [24] Mika, P.: Ontologies Are US: A Unified Model of Social Networks and Semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
- [25] Lin, X., Guo, L., Zhao, Y.E.: Tag-based social interest discovery. In: Proceeding of the 17th International World Wide Web Conference (2008)
- [26] Sibson, R.: SLINK: An optimally efficient algorithm for single-link cluster method. *Computer Journal* 16(1), 30–34 (1973)

Bi-clustering Gene Expression Data Using Co-similarity

Syed Fawad Hussain

Ghulam Ishaq Khan Institute of Engineering Sciences
and Technology, Pakistan
fawadsyed@gmail.com

Abstract. We propose a new framework for bi-clustering gene expression data that is based on the notion of co-similarity between genes and samples. Our work is based on a co-similarity based framework that iteratively learns similarity between rows using similarity between columns and vice-versa in a matrix. The underlying concept, which is usually referred to as bi-clustering in the domain of bioinformatics, aims to find groupings of the feature set that exhibit similar behavior across sample subsets. The algorithm has previously been shown to work well for document clustering in a sparse matrix representation. We propose a variation of the method suited for analyzing data that is represented as a dense matrix and is non-homogenous as is the case in gene expression. Our experiments show that, with the proposed variations, the method is well suited for finding bi-clusters with high degree of homogeneity and we provide empirical results on real world cancer datasets.

Keywords: Gene Expression Analysis, Bi-clustering, Co-similarity.

1 Introduction

The widespread use of microarray technologies during the last decade have enabled researchers to measure expression level of a large number (typically thousands) of genes under a number (typically in the hundreds) of different experimental samples (conditions). The resulting data is often represented as a matrix, called a gene expression data matrix, with rows usually representing an experimental condition, columns usually representing a gene and each cell of the matrix represents the intensity level of a given gene under a particular experimental condition. Each entry of the matrix corresponds to a numeric representation of the expression or activity of a particular gene under a given experimental condition, generally called sample. Applications of microarrays are quite wide, for instance the study of gene expression in yeast under different environmental stress conditions or the comparisons of gene expression profiles for tumors from cancer patients in order to find groups of genes that may exhibit similar patterns.

One usual goal in the analysis of such matrices is to extract the gene expression patterns inherent in the data and thus find potentially co-regulated genes. By comparing gene expression in normal and diseased cells, microarrays may be used to identify disease genes and targets for therapeutic drugs.

Many unsupervised learning techniques such as self organizing maps (SOM) [1], k-means [2], and hierarchical clustering [3] have been used for gene expression analysis. Clustering algorithms have proved useful for grouping together genes with similar functions based on gene expression patterns under various conditions or across different tissue samples. All of the above work is focused on clustering genes using conditions as features. These techniques, however, may fail to discover patterns if a cluster of features are active in only a subset of samples.

In the last decade a new group of algorithms, known as bi-clustering algorithms, that tends to find patterns in gene expression across only a subset of conditions have been widely used [4–8]. An example of a bi-partition would be $\{a; b; c\}$, and $\{d; e\}$ for objects, and $\{1; 4; 5\}$, $\{2; 3\}$, for features. This bi-clustering indicates that the commonality of objects from the partition $\{a; b; c\}$, is that they tend to share similar values among the feature group $\{1; 4; 5\}$. Similarly, features in $\{2; 3\}$ can be used to characterize objects in $\{d; e\}$. Bi-clustering algorithms have been well studied in the context of gene expression data analysis because it provides valuable information about putative regulation mechanisms and biological functions since they are good in finding local patterns and have been shown to outperform traditional clustering algorithms in bioinformatics, for instance see [9]. Similarly, several bi-clustering algorithms have been proposed in other domains, such as in text mining [10], [11], in social networking[12], etc. Such algorithms have, however, not been tested for the task of bi-clustering genes and vice versa.

One such approach to bi-clustering has been recently proposed by [13], where the authors have used the concept of co-similarity to produce bi-clusters for solving a similar problem in text mining. Their algorithm, named χ -Sim, utilizes the concept of higher-order correlations between words and documents to generate two similarity matrices, each built on the basis of the other. The concept of ‘higher-order’ co-occurrences has been investigated in [14] among others, as a measure of semantic relationship between words. The motivation behind χ -Sim is the exploitation of the dual nature of problem i.e. the relationship between groups of words that occur in a group of documents. Thus, documents are considered similar and hence grouped together, if they contain similar words and words in turn are considered similar and therefore grouped together, if they occur in similar documents. The concept can be expressed in the form of iterative mathematical equations which can then be solved to arrive at a solution.

Our work in this paper is motivated by the work of [13]. We feel that the domain of text clustering and gene expression analysis have significant similarity and thus, algorithms developed for one domain can be exploited, albeit with modifications, into the other. The χ -Sim algorithm is well suited for data coming from a text corpus that is homogenous and usually discrete as occurs with textual data represented using the Vector Space Model (VSM) [15]. This, however, is not usually the case for gene expression data where different intensity levels might be present across experiments and between different genes. To this end, we employ several pre-processing techniques and modify the algorithm proposed by [13] necessitated by the nature of our data.

The rest of the paper is organized as follows. Section 2 introduces the basic concept of the χ -Sim algorithm as described by [13]. In section 2.2, we highlight the

potential shortcomings of using χ -Sim on gene expression data and proposes pre-processing and modification steps in the algorithm. Detailed empirical results substantiating the usefulness of co-clustering are provided in Section 3. In section 4, we give a brief survey of the related work. Finally we conclude with a summary of our work and provide directions in Section 5.

2 The χ -Sim Algorithm

Throughout this paper we use the classical notation: matrices (in capital letters) and vectors (in small letters) are in bold.

Let \mathbf{D} be the data matrix representing a corpus having r rows and c columns; D_{ij} corresponds to the intensity of the j^{th} gene in the i^{th} sample. \mathbf{d}_i represents the row vector corresponding to gene i and \mathbf{d}^j represents the column vector corresponding to sample j . \mathbf{D}^T and \mathbf{d}_i^T denote the transpose of the matrix \mathbf{D} and document vector \mathbf{d}_i respectively. \mathbf{SR} and \mathbf{SC} represent the square and symmetric matrices of similarity between rows and similarity between columns of sizes $r \times r$ and $c \times c$ respectively with $SR_{ij} \in [0,1]$, $1 \leq i, j \leq r$ and $SC_{ij} \in [0,1]$, $1 \leq i, j \leq c$. $\mathbf{A} * \mathbf{B}$ represents the matrix multiplication between two matrices \mathbf{A} and \mathbf{B} while $\mathbf{A} \otimes \mathbf{B}$ denotes their Hadamard product.

2.1 The Algorithm

The χ -Sim algorithm is a co-similarity based approach which builds on the idea of generating simultaneously the similarity matrices between genes and between samples, each of them iteratively built on the basis of the other. We describe here the similarity between two genes (the similarity between samples being symmetrical). To calculate the similarity between two genes i and j , in addition to comparing the samples shared between the two genes, we also compare their ‘similar’ samples.

Thus all samples in document \mathbf{d}_i are compared to all samples in document \mathbf{d}_j . The product is defined as a measure of similarity similar to the cosine measure. However, when comparing samples with different indices, say D_{ik} and D_{jl} , their product is weighted by the similarity value between the samples k and l given by SC_{kl} .

Mathematically speaking, the similarity measure is given by $SR_{i,j} = \mathbf{d}_i * \mathbf{SC} * \mathbf{d}_j^T$ where \mathbf{d}_j^T denotes the transpose of the vector \mathbf{d}_j and the symbol ‘*’ denotes a matrix multiplication. The algorithm starts with two matrices \mathbf{SR} and \mathbf{SC} initialized to the identity matrix \mathbf{I} . In the absence of any prior knowledge about the similarity between any pair of genes, only the similarity value between a gene (or condition) with itself is considered as maximal and all other values are put to zero. \mathbf{SR} and \mathbf{SC} are then iteratively computed each one based on the other. Thus, genes are termed similar if they share similar conditions. Conditions, in turn are considered similar if they are up or down regulated in similar genes. This is termed as a co-similarity approach (as opposed to co-clustering which form hard clusters of the genes and conditions).

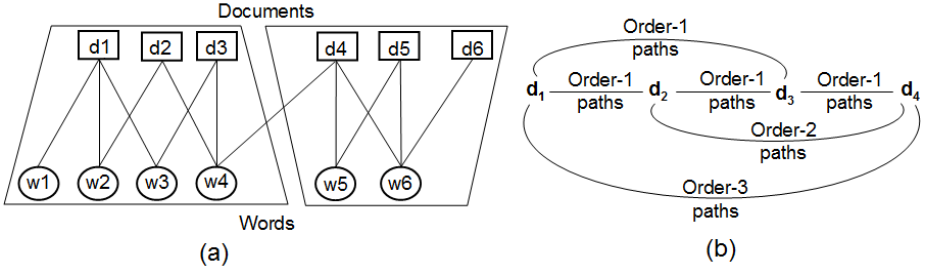


Fig. 1. (a) A bi-partite graph view of the matrix \mathbf{D} . The square vertices represent genes and the rounded vertices represent samples, and (b) some of the higher order co-occurrences between genes in the bi-partite graph.

We now present a graph theoretical interpretation of the algorithm which would enable us to better understand the working of the algorithm. Consider the bi-partite graph representation of a data matrix in Fig. 1(a) having 6 genes d_1 - d_6 and 6 conditions w_1 - w_6 . The genes and samples are represented by rectangular and oval nodes respectively and an edge between a gene i and a condition j in the graph corresponds to the entry D_{ij} in the matrix. There is only one order-1 path between genes d_1 and d_2 given by $d_1 \xrightarrow{D_{12}} w_2 \xrightarrow{D_{22}} d_2$. Hence the similarity value SR_{12} is given by the product $D_{12}D_{22}$. Note that since the \mathbf{SC} matrix is initialized as identity, at the first iteration, SR_{12} corresponds to the dot product between \mathbf{d}_1 and \mathbf{d}_2 (since $SC_{kl}=0$ for all $k \neq l$). The matrix $\mathbf{SR}^{(1)} = \mathbf{D} * \mathbf{D}^T$ thus represents the order-1 paths between all pair of genes \mathbf{d}_i and \mathbf{d}_j , $i=1..r$ and $j=1..r$. Each element of $\mathbf{SR}^{(1)}$ and $\mathbf{SC}^{(1)}$, denoted by $SR_{ij}^{(1)}$ and $SC_{ij}^{(1)}$ respectively is given by

$$SC_{ij}^{(1)} = \sum_{k=1}^r D_{ik} D_{kj} \quad \forall i, j \in [1, c] \quad (1)$$

$$SR_{ij}^{(1)} = \sum_{k=1}^c D_{ik} D_{kj} \quad \forall i, j \in [1, r] \quad (2)$$

Genes d_1 and d_4 do not have an order one path but are linked by d_2 and d_3 . The similarity value contributed via the document d_2 can be explicitly represented as $d_1 \xrightarrow{D_{12}} w_2 \xrightarrow{D_{22}} d_2 \xrightarrow{D_{24}} w_4 \xrightarrow{D_{44}} d_4$. From Eq. (1), the sub-path $w_2 \rightarrow d_2 \rightarrow w_4$ can be represented as $SC_{24}^{(1)}$, which is the first order path between w_2 and w_4 given by $SC^{(1)}$, and the contribution of d_2 in the similarity of $SR_{14}^{(1)}$ via d_2 can be written as $D_{12} * SC_{24}^{(1)} * D_{44}$.

This is a partial similarity measure as d_2 is not the only gene that forms a link between d_1 and d_4 . The similarity via d_3 (see fig 1(b)) is given by $D_{13} SC_{34}^{(1)} D_{44}$. The full similarity measure between d_1 and d_4 is thus given by $D_{12} SC_{24}^{(1)} D_{44} + D_{13} SC_{34}^{(1)} D_{44}$. This similarity can be represented as $\mathbf{SR}^{(2)} = \mathbf{D} * \mathbf{SC}^{(1)} * \mathbf{D}^T$ which corresponds to the \mathbf{SR} matrix at the second iteration. Hence similarity matrix $\mathbf{SR}^{(2)}$ at the second iteration corresponds to paths of order-2 (the similarity matrix between samples is similarly

given by $\mathbf{SC}^{(2)} = \mathbf{D}^T * \mathbf{SR}^{(1)} * \mathbf{D}$). Computing similarities this way, however, can result in an unbalanced scale since different vectors can have different lengths. Therefore, we normalize the values \mathbf{SR}_{ij} by $|\mathbf{d}_i| * |\mathbf{d}_j|$ where $|\mathbf{d}_i| = \sum_{k=1..c} (\mathbf{D}_{ik})$ and $|\mathbf{d}_j| = \sum_{k=1..c} (\mathbf{D}_{jk})$. Hence the generalized algorithm to find similarity measure at iteration n is given by:

Algorithm χ -Sim

Input: Document by term matrix \mathbf{D} , No. of iterations n ;

Step 1: Initialize matrices $\mathbf{SR}^{(0)} = \mathbf{I}$, $\mathbf{SC}^{(0)} = \mathbf{I}$

Step 2: for $t=1$ to n

$$\mathbf{SC}^{(t)} = \mathbf{D}^T \cdot \mathbf{SR}^{(t-1)} \cdot \mathbf{D} \otimes \mathbf{NC}$$

with $\mathbf{NC}_{i,j} = 1 / (|\mathbf{d}_i| \cdot |\mathbf{d}_j|)$ (3)

$$\mathbf{SR}^{(t)} = \mathbf{D} \cdot \mathbf{SC}^{(t-1)} \cdot \mathbf{D}^T \otimes \mathbf{NR}$$

with $\mathbf{NR}_{i,j} = 1 / (|\mathbf{d}_i| \cdot |\mathbf{d}_j|)$ (4)

Set diagonal of $\mathbf{SR}^{(t)}$ and $\mathbf{SC}^{(t)}$ to 1

Note that as a result of the L1 normalization used, the values of the matrix \mathbf{SR} (or \mathbf{SC}) are always bounded between 0 and 1 with 0 signifying no similarity (no connectedness between the two objects in order n paths in the graph) and 1 signifying maximum similarity.

2.2 χ -Sim as a Bi-clustering Algorithm

As mentioned previously, bi-clustering is the simultaneously clustering of rows and columns to identify row clusters that have some correlation with a certain column clusters. In this section, we describe how χ -Sim can be used as an algorithm to discover bi-clusters in non-homogeneous data such as gene expression data.

χ -Sim generates two similarity matrices – the row similarity matrix, \mathbf{SR} , and the column similarity matrix, \mathbf{SC} . The bi-clustering effect is achieved since each the similarity matrices, \mathbf{SR} and \mathbf{SC} , is built on the basis on the other, thus implicitly taking into account a feature selection. Applying the χ -Sim algorithm for bi-clustering gene expression data directly, however, poses a problem in performing comparisons. Genes, for instance, have different intensity profiles. For instance, if one gene’s intensity ranges from 1-50 while another genes intensity ranges between 10000-20000, then comparing these genes can result in rather skewed and meaningless similarities, particularly since the basic operation of χ -Sim is the dot product. Thus, genes having higher intensities will generate greater similarity, which might lead to undue and undesired high similarity values.

Data transformation, sometimes referred to as normalization or standardization, is necessary to adjust individual hybridization intensities of genes profiles such that intensity values between different genes are balanced and a comparison between them becomes meaningful [6], [16]. Transformation of raw data is considered an essential element of data mining since the variance of a variable determines the importance of that feature. We consider two types of transformation as discussed below:

Column Standardization. This is a classical technique in data analysis usually referred to as *centering* or *scaling*. Column Standardization (CS) involves taking the

difference between intensity values of a gene from the mean in units of standard deviation. Mathematically speaking, column standardization is defined as

$$D_{ij} = \frac{D_{ij} - \bar{\mu}_j}{\sigma_j}, \quad \forall i \in 1..m, j = 1..n \quad (5)$$

Where $\bar{\mu}_j = \frac{1}{m} \sum_{i=1}^m D_{ij}$ and $\sigma_j = \sqrt{\frac{1}{m} \sum_{i=1}^m (D_{ij} - \bar{\mu}_j)^2}$.

Row Standardization. Similarly, Row Standardization (RS) is defined as

$$D_{ij} = \frac{D_{ij} - \bar{\mu}_i}{\sigma_i}, \quad \forall i \in 1..m, j = 1..n \quad (6)$$

Where $\bar{\mu}_i = \frac{1}{n} \sum_{j=1}^n D_{ij}$ and $\sigma_i = \sqrt{\frac{1}{n} \sum_{j=1}^n (D_{ij} - \bar{\mu}_i)^2}$.

Usually, one needs to perform either row standardization or column standardization in order to transform the data. In some cases, row standardization followed by column standardization or bi-normalization might be needed. However, we are in a case where we need to compare pair of rows in order to determine the similarity between columns, and compare pair of columns so as to compute the similarity between rows as given in equations (3) and (4) respectively. Therefore, using either of equations (5) or (6) will only solve our problem partially.

As a result, we propose using both types of transformations given in (5) and (6) while maintaining two sets of the original dataset, \mathbf{D} . Let a row normalized matrix, transformed using equation (5), be denoted as \mathbf{D}_R while a column normalized matrix, transformed using equation (6), be denoted as \mathbf{D}_C . Then, equations (3) and (4) can be re-written as

$$\mathbf{SC}^{(0)} = (\mathbf{D}_R)^T \cdot \mathbf{SR}^{(t-1)}. (\mathbf{D}_R) \otimes \mathbf{NC} \text{ with } \mathbf{NC}_{ij} = 1/(\mathbf{d}_i \cdot \mathbf{d}_j) \quad (7)$$

$$\mathbf{SR}^{(0)} = (\mathbf{D}_C) \cdot \mathbf{SC}^{(t-1)}. (\mathbf{D}_C)^T \otimes \mathbf{NR} \text{ with } \mathbf{NR}_{ij} = 1/(\mathbf{d}_i \cdot \mathbf{d}_j) \quad (8)$$

Thus, by modifying the system of equations, we can now compare pair of rows and pair of columns using the appropriate normalized matrices. Of course, the overhead is that now we have to maintain separate copies of the dataset. The rest of the algorithm is unchanged as described previously in section 2.1. We will refer to this modified version of χ -Sim as χ -SIM_{mod}.

3 Experimentation

In order to validate the quality of bi-clusters generated by our algorithm, we used 2 real datasets that have been widely used in the literature and are publicly available.

3.1 Experimental Methodology

Colon Cancer. This dataset contains expression levels for 6500 human genes across 62 samples used by Alon et al [17]. The dataset corresponds to Colon Adenocarcinoma specimen collected from several patients, while normal tissues were also obtained from some of these patients. We selected the top 2000 genes with highest intensity across the samples. The resulting dataset contains 2000 genes and across 40 tumorous and 20 normal colon tissues. Note that this dataset do not contain negative values and only 1909 of the 2000 genes are unique. We further preprocessed the data by removing genes with $\text{lmax}/\text{minl} < 15$ and $\text{lmax} - \text{minl} < 500$ leaving a total of 1096 genes.

Leukemia. This dataset was used by Golub et al. [18] and contains 7129 genes across 72 samples. The dataset corresponds to RNA extracted from bone marrow samples of patients with leukemia at the time of diagnosis. About 47 samples were suffering from Acute Lymphoblastic Leukemia (ALL) while 25 samples were suffering from Acute Myeloid Leukemia (AML). We first used a floor value of 100 and a ceil value of 16000. Only genes with $\text{lmax}/\text{minl} < 5$ and $\text{lmax} - \text{minl} < 500$ were selected leaving a total of 3571 genes. The preprocessing steps applied here have been used to enable direct comparison with the results of Cho et al. [6].

The experimentation was run as follows: Each of the dataset was taken and the preprocessing was applied, resulting in a reduced matrix with only the selected genes across all the samples. The modified χ -Sim algorithm was run on the data matrix using Matlab and Agglomerative Hierarchical Clustering using Wards linkage was applied to the resulting similarity matrices. The number of sample clusters was set as the real number of clusters whereas the number of gene clusters was set to 100 (as in [6]). The top k bi-clusters were chosen as gene clusters with the greatest homogeneity across the sample clusters.

3.2 Sample Cluster Analysis

Our first analysis attempts to verify the quality of the sample clusters. We take the sample clustering from the generated bi-clusters and evaluate the accuracy of the samples. The accuracy measures the number of samples that were correctly grouped together in one bi-cluster as a percentage of the total samples in the dataset. For instance, the two sample categories for the colon cancer dataset are those samples that have tumor and those that do not have tumor. We report the results based on

1. applying no transformation (pre-processing) at all; and
2. Applying both row and column transformation (as discussed in the previous section; see equations (7) and (8))

The results are reported in table 1 below.

Table 1. Accuracy of sample clustering in the bi-clusters

	NT	RS + CS
Colon	0.8871	0.9032
leukemia	0.9583	0.9722

As can be seen from the table, without any transformation (i.e. baseline χ -Sim), only 88.71% of the samples are correctly clustered together. However, when applying the χ -SIM_{mod} algorithm, the accuracy of the sample clusters rises to 90.32%. A similar increase in the accuracy is observed in the case of leukemia dataset.

3.3 Gene Cluster Analysis

Unlike the condition clusters, we do not have a priori knowledge of the gene clusters. One way to analyze the gene clusters is by visually analyzing the profiles of genes that are clustered together in the bi-clusters. Ideally, we would like genes clustered together to exhibit similar profile behaviors under the condition clusters. Thus, plotting the bi-clusters give us a visual reference to the “profile” of the bi-cluster and the constituent genes. This approach has been employed previously to judge the quality of the generated bi-clusters, for instance by [5], [6], [17] among many others.

In order to report the best bi-clusters, we generated several bi-clusters and report the top k bi-clusters that exhibit similar profiles amongst the samples. The result for the top 2 clusters (k=2) for colon cancer dataset and leukemia dataset is shown in Figure 2. The x-axis corresponds to the tissues while the y-axis shows the intensity level of the gene. The figure clearly illustrates that χ -SIM_{mod} captured homogenous gene expression patterns in the gene clusters. Two bi-clusters representing a healthy tissue and tumor tissues for colon cancer dataset are shown in figure 2(a) and (b)

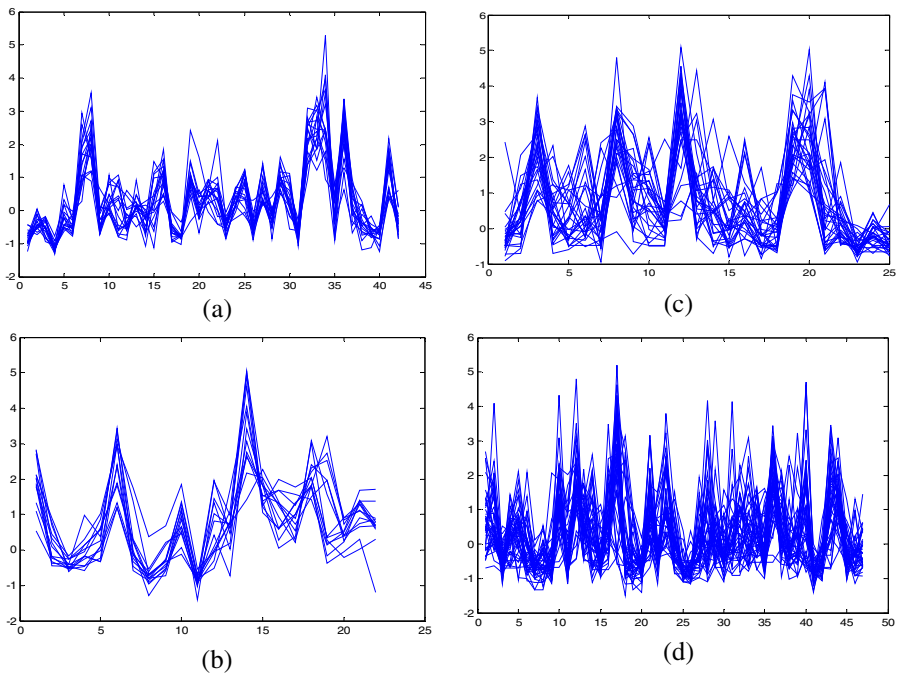


Fig. 2. Top 2 bi-clusters generated by applying the modified χ -Sim algorithm on (a) colon cancer dataset of tumor tissues, (b) colon cancer dataset of normal tissues; and (c) leukemia cancer dataset with AML, (d) leukemia dataset with MLL.

respectively. As mentioned in [17] and [6], ribosomal genes usually have higher intensities in tumor samples and comparatively lower intensities in non-tumor (normal) samples. In fact, most of the ribosomal protein genes given in [17] can be found in two clusters.

Similarly, for the leukemia dataset, Figure 2(c) represent bi-cluster corresponding to genes profiles of AML while figure 2(d) corresponds to gene profiles of MLL. We can see that χ -SIM_{mod} is able to discover bi-clusters that have several genes behaving similarly over a large number of experimental conditions. For instance the bi-cluster shown in figure 2(a) contains only 16 genes showing similar behavior across all 42 conditions. Similarly, we also discover bi-clusters have many genes showing similar characteristics across different conditions, for instance in figure 2(c) where 27 genes show similar behavior across 25 conditions.

4 Related Work

Several algorithms for clustering of gene expression data have been proposed in the literature most algorithms were designed keeping bi-clustering of gene expression data in mind. Several kinds of bi-clusters have been identified and algorithms proposed that are able to identify them (see for instance [19] for common bi-clusters and a survey of algorithms proposed in the literature.). Perhaps the earliest and most well-known algorithm for finding bi-clusters in gene expression data was proposed by [5] that finds sub-matrices with the minimal squared residue. This algorithm however uses a greedy approach to find bi-clusters and does not take the overall similarity between genes and samples into account.

Tanay et al. [20] proposed an algorithm, known as the Statistical-Algorithmic Method for Bi-cluster Analysis (SAMBA) to discover bi-clusters. The model is based on a data matrix corresponding to a bipartite graph and uses statistical models to solve the problem by identifying bi-cliques in the graph. Depending upon whether a gene is up-regulated or down regulated, the corresponding edges are assigned a weight. The Order Preserving Sub Matrix (OPSM) technique proposed of Ben Dor et al. [21] assumes a probabilistic model of the data matrix. They define a bi-cluster as a group of rows such that the expression values in all features increase or decrease simultaneously.

The Bimax algorithm was used as a reference method in a comparative study of bi-clustering algorithms by [4]. The algorithm uses on 0's and 1's which can be obtained by discretization as a prior preprocessing step. A bi-cluster is then defined as a sub-matrix containing all 1's i.e. a set of genes that are up-regulated in a set of conditions.

More recently, a Minimum Sum Squared Residue Co-Clustering algorithm was proposed by Cho et al. [6] that do not take into account a correspondence between row and column clusters as such, but consider sub-matrices formed by them with the overall aim to minimize the sum of squared residue within the sub-matrix. Cho et al. proposed an algorithm that is based on algebraic properties of a matrix. The algorithm runs in an iterative fashion and on each iteration, a current co-clustering is updated

such that sum of squared residue is not increased. In other samples, the algorithm monotonically decreases and converges towards a locally optimal solution.

5 Conclusion

We provide an extension and adaptation of the χ -Sim algorithm for bi-clustering gene expression datasets. It has been successfully demonstrated that the proposed χ -Sim algorithm does perform bi-clustering and its results are better than other contemporary traditional techniques. The quality of the results was verified by bi-clustering several publicly available cancer data sets, and analyzed the results of both the gene and sample clusters.

Our work is also significant in that we have used an algorithm that was fundamentally developed for text clustering and adapted it to perform bi-clustering of gene expression data. This is an interesting scenario and one needs to further investigate if other co-clustering algorithms can be adapted for bi-clustering problem and vice versa and this will form a major part of our future work.

References

- [1] Tamayo, P., et al.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America* 96(6), 2907 (1999)
- [2] Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. *Nature Genetics* 22, 281–285 (1999)
- [3] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95(25), 14863 (1998)
- [4] Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., Zitzler, E.: BicAT: a biclustering analysis toolbox, vol. 22. Oxford Univ. Press (2006)
- [5] Cheng, Y., Church, G.M.: Biclustering of expression data, pp. 93–103 (2000)
- [6] Cho, H., Dhillon, I.S.: Coclustering of human cancer microarrays using minimum squared residue coclustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 385–400 (2008)
- [7] Wee-Chung Liew, A., Law, N.F., Yan, H.: Recent Patents on Biclustering Algorithms for Gene Expression Data Analysis. *Recent Patents on DNA & Gene Sequences* 5(2), 117–125 (2011)
- [8] Gu, J., Liu, J.: Bayesian biclustering of gene expression data. *BMC Genomics* 9(1), S4 (2008)
- [9] Prelic, A., et al.: A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22(9), 1122–1129 (2006)
- [10] Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., Modha, D.S.: A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 509–514 (2004)

- [11] Hussain, S.F., Bisson, G., Grimal, C.: An improved co-similarity measure for document clustering. In: Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 190–197 (2010)
- [12] Giannakidou, E., Koutsonikola, V., Vakali, A., Kompatsiaris, Y.: Co-clustering tags and social data sources. In: The Ninth International Conference on Web-Age Information Management, pp. 317–324 (2008)
- [13] Bisson, G., Hussain, F.: Chi-Sim: A New Similarity Measure for the Co-clustering Task. In: International Conference on Machine Learning and Applications, pp. 211–217 (2008)
- [14] Lemaire, B., Denhière, G.: Effects of high-order co-occurrences on word semantic similarities, Arxiv preprint arXiv:0804.0143 (2008)
- [15] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Communications of the ACM* 18(11), 620 (1975)
- [16] Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* 97(457), 77–88 (2002)
- [17] Alon, U., et al.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* 96(12), 6745 (1999)
- [18] Golub, T.R., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439), 531 (1999)
- [19] Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. *IEEE Transactions on computational Biology and Bioinformatics*, 24–45 (2004)
- [20] Tanay, A., Sharon, R., Shamir, R.: Biclustering gene expression data. In: International Conference on Intelligent Systems for Molecular Biology (2002)
- [21] Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology* 10, 373–384

CCE: A Chinese Concept Encyclopedia Incorporating the Expert-Edited Chinese Concept Dictionary with Online Cyclopedias

Jiazhen Nian, Shan Jiang, Congrui Huang, and Yan Zhang*

Department of Machine Intelligence,
Peking University,
Beijing 100871, P.R. China
{nian, jsh, hcr}@pku.edu.cn,
zhy@cis.pku.edu.cn

Abstract. Bag-of-words is the most common-used method in text mining tasks and many other applications. However, this method has some obvious shortcomings, such as ignoring semantic information. While in document analysis, semantic information always plays a more important role than individual words. To tackle this problem, we need to borrow semantic information from ontologies to learn the text information better. An expert-edited ontology is usually well structured and is more authoritative than an online cyclopedia. On the other hand, due to the costly editing, it is rather difficult for expert-edited ontologies to keep up with a deluge of new words. In this paper, we propose a method to construct a Chinese ontology to keep the carefully-designed structure of an expert-edited ontology, meanwhile embody new vocabulary from an online cyclopedia. We name the enhanced ontology as Chinese Concept Encyclopedia (CCE) and employ it in some text mining applications. The experimental results show that CCE outperforms the expert-edited ontology Chinese Concept Dictionary (CCD).

Keywords: CCD, CCE, Ontology Integration.

1 Introduction

With the development of the Internet, the era of great explosion of information has already come. Knowledge discovery and data mining are useful tools to ravel out information blast, and semantic information will definitely be very helpful to understand documents better. To learn to get the semantic information, adequate background knowledge is necessary. And ontologies always play the role of the provider of background knowledge in document analysis.

Chinese words are formed by individual pictographic characters, and usually contain two or three characters. Twenty thousand characters construct more than ten million Chinese words. Bag-of-words method always only gets the

* Corresponding author.

surface meaning of words but is powerless to get the deeper level of semantic information.

Researchers have made great effort for semantic repository construction and maintenance. There are plenty of Chinese ontologies available, such as HowNet [1] and Chinese Concept Dictionary (CCD) [3]. CCD has a similar structure with WordNet and is widely used in text mining. The basic component of CCD is concept. A concept is represented by a synset, and usually with a gloss. Between concepts, various kinds of semantic relations hold, such as antonymy, hyponymy and holonymy, etc.

Although CCD has many good properties, it is rather static and its capability of containing new words is poor compared with online cyclopedias. With the springing up of new words, the costly way of manual editing and maintenance is sometimes powerless. Thus, enriching CCD with information extracted from online cyclopedias is worthwhile and beneficial.

As a product of web 2.0, the collaborative way of editing and the huge quantity of editors makes Baidu Baike the most famous Chinese on-line cyclopedia. According to the statistics, there are over 3000 new entries being added to Baidu Baike, and about 10,000 entries are edited every day. By the year 2011, Baidu Baike included 3,439,372 entries, with about 1 million created before 2008 [4]. We believe that integrating CCD with Baidu Baike will be rewarding. In this paper, we propose a method to construct a new ontology named as Chinese Concept Encyclopedia (CCE). CCE keeps the carefully-designed structure of CCD and has the character of a dynamic capability of containing new words due to the information extracted from Baidu Baike as well. As other semantic repositories, CCE can be applied in many applications such as query reformulation, text mining, web page ranking, etc.

Because of the collaborative way of editing, the hierarchy of Baidu Baike is not as authoritative as that of an expert-edited ontology. However, the category information provides important clues for the integration. The descriptions and the tags marked by users of the entities in Baidu Baike also help us to add the entities into CCE.

As far as we know, this work is the first attempt to build a Chinese language knowledge database using Internet Cyclopedia.

The remainder of this paper is organized as follows. We review the related works in Section 2. The details of the integration of CCD and Baidu Baike are described in Section 3. The experimental results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Related Work

2.1 Chinese Concept Dictionary

At present time, Natural Language Processing (NLP) is focusing on the processing of content information, of which a WordNet-like Chinese Concept Dictionary

¹ <http://baike.baidu.com/view/1.htm>

can be considered as an important building block. With the development of NLP, semantic ontologies become more and more important.

Chinese Concept Dictionary (CCD) is developed by Institute of Computational Linguistics, Peking University. It is a WordNet-like semantic lexicon of contemporary Chinese and is structured from the viewpoint of Computational Lexicography [3]. CCD is bilingual and is compatible with WordNet, but it does not mean that CCD is same with WordNet. As a matter of fact, there are differences between Chinese and English in the organization structure and processing method.

In CCD, concepts are used as basic units. Concepts are defined as synsets and a variety of relations are defined between synsets, such as holonymy and hyponymy. Labeled tree is used as the basic description of hyponymy hierarchy. The basic concept categories are the 25 categories in the fundamental level of WordNet. And inferior categories are derived from the superior ones. All the attributes of words are summarized from Chinese Grammar Information Dictionary, which is a traditional Chinese dictionary.

CCD contains more than 66,000 noun concepts, 12,000 verbs and 21,000 concepts of adjunct word. As an important Chinese semantic repository, CCD is widely used in many applications.

2.2 English Concept Cyclopedia Enhancement

WordNet is one of the most widely-used lexical databases for English. Many researchers have already proposed numbers of methods to enhance WordNet.

Kevin Knight and Steve K. Luk propose a method to enhance WordNet to use it in knowledge-based machine translation (KBMT) system. They merge more than 70,000 various online dictionaries, semantic networks, and bilingual resources, and construct a new ontology [4].

For the shortage of not expressing the distinction between compatible versus incompatible co-hyponyms and polysemy versus homonymy, Sara Mendes and Rui Pedro Chaves enrich WordNet with qualia information. Such an enrichment is to take place, so what is semantically specified for a hyperonym is shared by its hyponyms, keeping synsets as simple as possible [7].

Wikipedia² as a well-known on-line cyclopedia has drawn a lot of attention from academic researchers in recent years. Jiang et al construct an enhanced ontology, WorkiNet [2], by integrating the information of Wikipedia into WordNet. This ontology is the combination of expert-edited ontology and web cyclopedia. The performance of WorkiNet in text mining is competitive.

Many algorithms are developed to harvest lexical resources, but few organize the mined terms into taxonomies. According to Zornitsa Kozareva's work, a semi-supervised algorithm [5] is proposed, that uses a root concept, a basic level concept, and recursive surface patterns to learn automatically from the Web hyponym-hypernym pairs subordinated to the root, meanwhile a Web based concept positioning procedure to validate the learned pairs' is-a relations is

² <http://www.wikipedia.org>

proposed. Comparing to WordNet, it discovers many additional ones lacking in WordNet.

Another work of Elisabeth Wolf is also aimed on Wikipedia. They propose a method to convert Wikipedia to a sense inventory [10]. An aligned sense inventory of both resources has two major benefits: the coverage of senses can be increased and enhanced information about aligned senses can be obtained. Sangno’s work examines and proposes the automatic generation of concept hierarchies using WordNet [6]. The automatic generation of hierarchies makes the generation process more efficient than an manually built one. A new ontology YAGO [8] is proposed by Fabian M. Suchanek It’s automatically derived from Wikipedia and WordNet, and it has high coverage and precision.

3 Construction Methodology for CCE

CCD, the edition of 2009, contains 66,590 concepts. As an expert-edited ontology, its capacity is to some extent very large. However, compared with the enormous new words emerging on the web, this number is not that compelling. To keep CCD fresh, we propose a method to merge entities of Internet Chinese Cyclopedias such as Baidu Baike into it automatically and construct a new ontology, named as CCE. For each entity, we use its category information as an important clue to figure out its position in the new ontology. If the entity is a polysemant word, it will be attached to all of its hypernyms.

3.1 Notation and Definition

Ontology. Concepts are basic units of CCD. Each one refers to a set of synonymous words. Various kinds of semantic relations exist between concepts, such as holonymy and hyponymy. We mainly focused on hyponymy relation in this paper.

Definition 1. Chinese Concept Dictionary concept: A CCD concept π is a two-tuples $\langle \text{SynSet}, \text{HypoSet} \rangle$, where SynSet is the synonym set of π , and HypoSet is its hyponym set. We refer to the items with $\pi.\text{SynSet}$ and $\pi.\text{HypoSet}$ respectively.

Entities are instantiations of a kind of ontology appearing on web. In this paper, we define an “Entity” as a word collected by Baidu Baike.

Definition 2. Baidu Baike Entity: A Baidu Baike Entity ϕ is a triple $\langle \text{term}, \text{disc}, \text{CateSet} \rangle$, where term denotes the title of ϕ , disc is the description of ϕ , CateSet is the set of categories which are tagged by users, and RelateSet is the set of related entities of ϕ . We refer to the items with $\phi.\text{term}$, $\phi.\text{disc}$, and $\phi.\text{CateSet}$ respectively.

For instance, ϕ , the article “Apple”³: $\phi.\text{term}$ is “apple”, $\phi.\text{disc}$ is “Apple is a kind of fruit...”, $\phi.\text{CateSet}$ is {“movie”, “brand”, “fruit”, “computer brand”,

³ <http://baike.baidu.com/view/1331.htm>

“Rosaceae”}. We make the ontology instantiation by building a semantic tree, whose node represents every entity. And nodes which represent semantic concepts are also the basic units of CCE.

Definition 3. Chinese Concept Encyclopedia Node: A node π is a tetrad $\langle SynSet, hype, HypoSet, depth \rangle$, where *SynSet* denotes the semantic concept of π , *hype* is the hypernym of π , and *HypoSet* is the set of its hyponym concepts, *depth* shows the node’s depth in the semantic tree.

For instance, π , the entity “network”⁴: $\pi.SynSet$ is “network”, $\pi.hype$ is “communication”, $\pi.HypoSet$ is {“web site”, “broadcast network”, “Internet”, etc.}, $\pi.depth$ is 7.

3.2 The Structure of Baidu Baike

Baidu Baike is the most famous Chinese online cyclopedia, which is collaboratively edited by millions of users. By Sept. 2010, Baidu Baike has already contained 2,559,745 entities, with about 1.5 million created in the recent two years.

Baidu Baike has a 3-level hierarchical structure. The top two levels are the words that can be used as categories. All the entities are categorized and then attached under the categories. The categories in the first level are “People”, “Culture”, “Art”, “History”, “Technique”, “Life”, “Geography”, “Society”, “Sport”, “Nature”, “Science” and “Economy”. Each main Category has several secondary classifications. Take “People” as an example. Its secondary classifications are “Scientist”, “Virtual characters” and other eight categories.

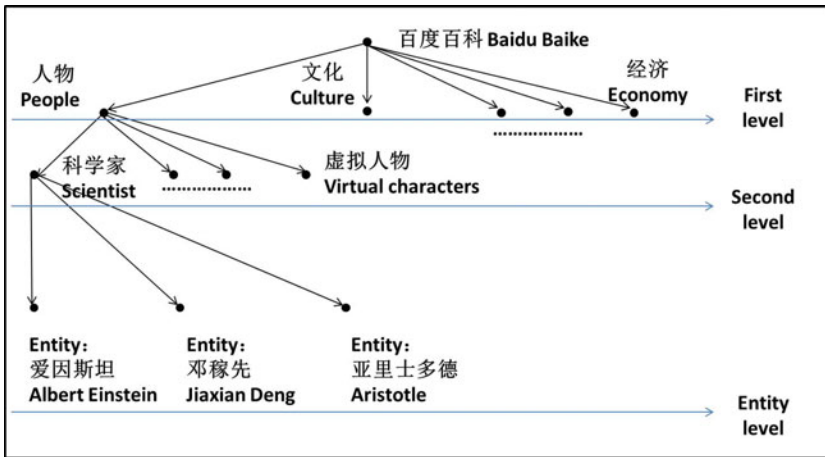


Fig. 1. Structure of Baidu Baike

⁴ <http://baike.baidu.com/view/3487.htm>

CCD is expert-edited, so the relation between entities is quite accurate. Baidu Baike classifies entities into different categories by giving tags for each entity. Most of the categories are tagged by users collaboratively. Ordinary users cannot do this job as accurately as experts, so redundancy and inaccuracy exist in Baidu Baike’s hierarchy. For example, the word “Inception” and “Movie” are both the hyponym of “entertainment”. As we know Inception is a part of movie, so in the new ontology, “Inception” is a hyponym of “movie”, meanwhile “movie” is a hyponym of “entertainment”. If this entity is in CCD, we may say “inception” is both “Movie” and “Entertainment”’s hyponym. When it comes to the construction of CCE, we will analyze “Inception”’s categories first. It is easy to find “Movie” is the hyponym of “Entertainment”, so we only put “Inception” into the HypoSet of “Movie”.

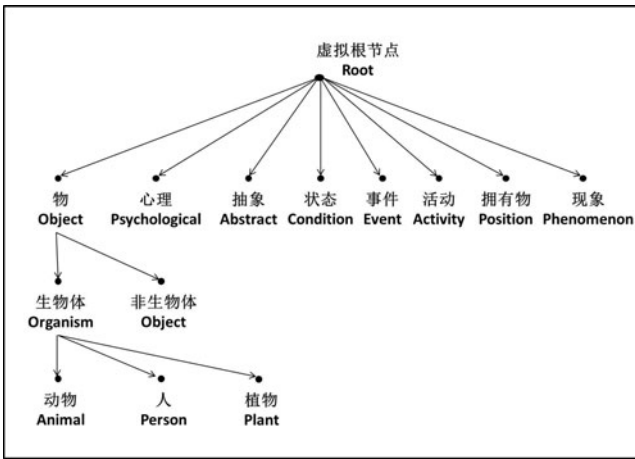


Fig. 2. Structure of Chinese Concept Dictionary

CCE inherits the structure of CCD. We set “things”, “Position” and other six categories under the root. And all the entities will be organized in a tree-like structure, where each node represents an entity.

3.3 Method for Ontology Generation

We will use the category information of an entity as an important clue to figure out its position. The main idea is to find a path from root to the subject entity. That is to say, this algorithm will figure out all the ancestors of the entity recursively.

The method can be divided into two steps: expanding category and integrating entity. In the first step, we expand the fundamental structure of CCD with the entities in the top two levels of Baidu Baike, which are elaborately designed and structured and are more static compared with other entities in Baidu Baike. In the second step, utilizing the category information, we find an appropriate position for each of the entities in the third level of Baidu Baike.

Expanding the Main Categories. As we know, all the Baidu Baike entities are in a three-level structure. In the expanding category step, we will use the entities in top two levels to enhance the fundamental structure of CCD.

All the 12 concepts of the first layer of Baidu Baike's categories have been in CCD. So we just need to add the second layer of the categories. We denote these Baidu Baike entities as an entity set Φ , and all the concepts in CCD as a set Π , which is the fundament of CCE. In this step, we need to find appropriate hypernym concept for each entity in the second layer of Baidu Baike, and put the entity to the hyponym set of its hypernym. To find the appropriate hypernym concept of an entity, we can use the category information as clues. Since the categories of Baidu Baike are manually edited by the editors, and a category can provide higher level abstraction of an entity, we believe that using categories as the hypernyms of entities in the second level of Baidu Baike is feasible. If an entity in the second level has more than one categories, we add the entity into the hyponym set of every concept containing any of the categories.

For instance, the entity "chemist" does not appear in CCD, but its superior entity "People" exists in CCD, so we put "Chemist" into the hyponym set of "People".

Algorithm of expanding the main categories;

foreach i, ϕ_i of Φ **do**

foreach j, ϕ_j of $\phi_i.CateSet$ **do**

 find π from Π where $\phi_j.term \in \pi.SynSet$;

 create a CCE node $\pi_n(\phi_i.term, \pi.entity, \emptyset, \pi.depth + 1)$;

 put π_n into $\pi.HypoSet$;

This step uses naïve method, however, this algorithm ensures most entities of Baidu Baike can find their hypernyms in the construction processing of CCE.

Integrating Entities into CCE. In the entity integrating step, our target is to figure out the appropriate position of entities appearing in the third level of Baidu Baike. Generally speaking, the categories can be still served as clues for finding the hypernyms. However, the situation is more complicated than that of the entities in the second level. We should first make sure whether it is a polysemant or not. If the answer is yes, we should select the most suitable categories as the entity's hypernyms.

First we download all the Baidu Baike entities and store them as an entity set. Then we extract all the entities which appear in category set of other entities and mark them as Φ_1 , the remaining ones that does not appear in the category of any entity are denoted as Φ_2 . Ψ_0 denotes the entity set of CCE. Then we add Φ_1 and Φ_2 into Ψ_0 respectively.

The iteration is to shift entity from Baidu Baike entity set to entity set of CCE until no one in Baidu Baike entity set can be moved into CCE entity set. In this step, the situation is more complicated than that in the expanding step because

the category information given by the editors is not as high quality as that of the entities in the second level of Baidu Baike. For each entity in Baidu Baike entity set, we check its category set, if there are more than one categories appear in the same path in CCE, we consider the one with the maximum depth as the candidate rather than the other shallower ones and attach the entity under the candidate node in order to avoid loops in CCE.

The main idea of this step is to recursively add entities from Baidu Baike into CCE that can find an appropriate position in CCE. When there is a new entity to be added into CCE, we hope that we can find the adding position as precisely as possible. So if more reasonable candidates for the adding position is provided, the integrating quality is more likely to be improved. And when an entity π_a is in the category set of another entity π_b , if we add π_a first, π_a will be an extra candidate for the adding position of π_b . Thus we add the entities appearing in the category set of others first, then we turn to the remaining entities until there is no entities in Baidu Baike that can be add into CCE.

For instance, the category set of the entity “Apple” is {“fruit”, “computer company”, “company”, “movie”, “French movie”, ...}. We find “company” is the ancestor node of “computer company” in CCE, so we delete “company” from this set. For the same reason, “movie” is deleted. Then we create CCE entity “Apple” and add this node under “computer company” and “French movie”. The concept “fruit” in CCD has the hypernym “apple”, so we do not add “apple” to “fruit” repeatedly.

Integrating Entities into CCE;

while *no more Baidu Baike entity is edited* **do**

foreach ϕ_i of Φ_1 **do**

foreach ϕ_p, ϕ_q of $\Phi_i.CateSet$ **do**

if ϕ_p is the hyponym of ϕ_q **then**

 └ Dislodge ϕ_p from $\Phi_1.CateSet$

foreach j, ϕ_j of $\phi_i.CateSet$ **do**

 find π from Π where $\phi_j.term \in \pi.SynSet$;

 create a CCE node $\pi_n(\phi_j.term, \pi.entity, \emptyset, \pi.depth + 1)$;

 put π_n into $\pi.HypoSet$;

foreach ϕ_i of Φ_2 **do**

foreach ϕ_p, ϕ_q of $\Phi_i.CateSet$ **do**

if ϕ_p is the hyponym of ϕ_q **then**

 └ Dislodge ϕ_p from $\Phi_1.CateSet$

foreach j, ϕ_j of $\phi_i.CateSet$ **do**

 find π from Π where $\phi_j.term \in \pi.SynSet$;

 create a CCE node $\pi_n(\phi_j.term, \pi.entity, \emptyset, \pi.depth + 1)$;

 put π_n into $\pi.HypoSet$;

3.4 Semantic Distances and Similarity

The depth of π is defined as follow:

$$depth(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is root,} \\ depth(\pi.Hypernyn) + 1 & \text{otherwise,} \end{cases} \quad (1)$$

The depth of the root is 0, for other concept, their depth depends on their parent node. The smaller the depth is, the more general the concept is.

The distance of two entities can be defined as the shortest path between them. That is the sum of the distances between them and their common ancestor entity.

The shortest path length between two concepts is defined as follows:

$$\begin{aligned} len(\pi_1, \pi_2) &= \min_{\pi^* \in \Pi} ((depth(\pi_1) - depth(\pi^*)) + (depth(\pi_2) - depth(\pi^*))) \\ &= \min_{\pi^* \in \Pi} (depth(\pi_1) + depth(\pi_2) - 2 \times depth(\pi^*)) \end{aligned} \quad (2)$$

π^* is the lowest super-ordinate (or most specific common subsumer) of π_1 and π_2 .

It is observed that the similarity between deeper concepts should be higher than the shallower ones which share the same shortest path length because the deeper ones are more specific while the shallower ones are more general [9]. Based on this observation, we define the semantic similarity between two concepts [2]:

$$sim(\pi_1, \pi_2) = \frac{\log \frac{len(\pi_1, \pi_2)}{depth(\pi_1) + depth(\pi_2)}}{\log(2(max_{\pi \in CCE} depth(\pi) + 1))} \quad (3)$$

4 Experiments

4.1 Experiment Setup

We download about 2,800,000 Baidu Baike entities in September 2010. And the edition of CCD that we use is that of Sept. 2009, which contains 66,590 Chinese words. Using the method described in Section 3, we add 1,791,620 individual entities from Baidu Baike and finally construct an new ontology CCE with more than 2,000,000 concepts.

In the experiments, we employ both CCD and CCE in document classification, document clustering, and blog comment analysis.

4.2 Case Study of CCE

To evaluate the structure and content information of CCE, we invite three volunteers, who make their own evaluation respectively.

We randomly select 100 nodes from CCE, which is added from Baidu Baike to CCD. Then the volunteers give each of them a score according to their semantic relations with their hypernyms. If an entity is added to an appropriate concept and the semantic relationship between them matches hyponymy perfectly, this

case will be put into Excellent. If the matching quality is a little lower, it will be put into Good. If the relation between the entity and the concept is not that strong, but still reasonable in some sense, it will be put into Fair. If the case is difficult to judge, we will put it into Neutral. Finally, the wrong cases are left to Bad. Each volunteer gives a score for every node, then we average all scores and round the average score to its nearest grade as the final result.

The results are shown in Table. 1.

Table 1. Case Study

Excellent	Good	Fair	Neutral	Bad
35%	22%	22%	9%	12%

And the results in different categories are presented in Table 2.

Table 2. Case Study in Specific Category

Categories	Amount	Excellent	Good	Fair	Neutral	Bad
Material object	59	44.07%	22.03%	20.34%	5.08%	8.48%
Physiology psychology andfeeling	20	20.00%	15.00%	20.00%	15.00%	30.00%
Knowledge and others	21	23.81%	28.57%	28.57%	14.29%	4.76%

We examine the cases in *Bad* and find that most cases are polysemants. According to the algorithm depicted in Section 3, when it comes to polysemants, we will add them to all of their superior entities. However, some superior entities can be considered as higher level abstraction of their son node but not the hypernyms of the words. Especially in the domain of physiology and psychology, there is usually big difference between the meanings of the same word. So the qualities between different domains of the new ontology are sometimes varied.

4.3 Document Classification

In this section, we apply CCE in document classification, and compare its performance with that of CCD.

Our data set is crawled from Sohu News⁵ from October to December in 2009. The documents are categorized into automobile, economy, education, health, IT, career, house, military and culture. In education, career and house, there are few documents and most of them are quite short. To guarantee data scale and documental validity, we chose automobile, economy, health, IT and military five categories. In each category, we sample 1000 documents randomly.

Then we use Naïve Bayes classification algorithm. And we use CCD and CCE as the semantic repository for term segmentation respectively. Then we get the following results.

⁵ <http://news.sohu.com>

To examine the effect of binary classification, we choose 300 documents from economy and health. We randomly select 270 documents as the training data and the remainder 30 documents are used for testing. And the results are shown in the second row as “binary”. In the third row “multicase”, we choose 300 documents from economy, health, automobile, IT and military. 270 documents are taken as training set as well. The results are given in Table 3.

Table 3. Precision Rate of Classification

	CCD	CCE
Binary	93.3%	96.7%
Multicase	86.7%	93.3%

We can draw the conclusion that CCE can improve the performance of classification. The reason is that CCE contains abundant vocabulary; especially new vocabulary appearing on the Internet. For the news containing more new words, CCE shows better performance.

4.4 Document Clustering

We use the dataset which is used in Section 4.3 and sample 20 articles in each category randomly.

We apply K-means algorithm with the distance between two vectors defined by ourselves which measures the semantic relationship between two document fragments.

First, we use CCD and CCE as the semantic repository for term segmentation respectively. Then we calculate the tf-idf of the word set of each article and and pick out the top 20 words with the maximum tf-idf value of each article.

Then we use tf-idf as the weight of each word and calculate the semantic distance between two article fragments by the following formula:

$$distance(Article_1, Article_2) = \sum_i \sum_j \log(sim(word_{1i}, word_{2j}) * tfidf_{word_{1i}} * tfidf_{word_{2j}}). \quad (4)$$

We define purity to evaluate the performance of clustering. Let $G = \{g_1, g_2, \dots\}$ denote the cluster set generated by K-Means, and $C = \{c_1, c_2, \dots\}$ denote the pre-defined category information of the documents, which can be used as the criterion for the clustering result, $D = \{d_1, d_2, \dots\}$ denote the set of documents. To compute the purity, we define the most frequent original category c_i appearing in g_k as the category c_i ranged to. Then the purity is measured by the number of correctly assigned documents divided by the number of documents. Formally:

$$Purity = \frac{1}{|D|} \sum_k \max_i |g_k \cap c_i|. \quad (5)$$

Finally we use k-means for clustering and compare the purity of clustering results.

We sample two experiment sets from data set randomly. For each sampling, we run three times, then calculate the average purity. The results are shown in Fig. 3.

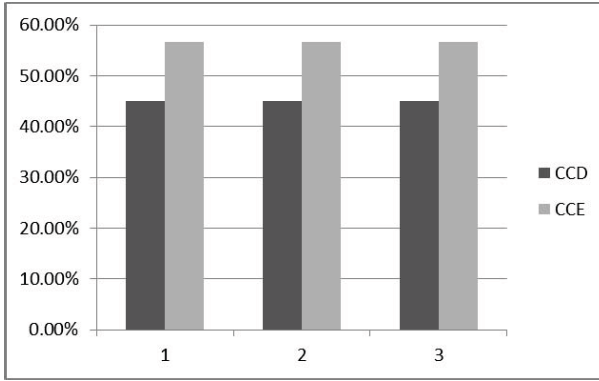


Fig. 3. Clustering purity

4.5 Relevance Analysis in BlogSphere

In blog system, a lots of comments will be posted by visitors at the end of blog contents. However, numbers of the comments do not have much relevance to the blogs. In this section, we propose an algorithm to analysis the relevance between comments and blogs.

We use 80 blog posts from Sina⁶, Baidu⁷, Sohu⁸ and IFeng blog⁹ web sites. For each blog, we randomly select 20 comments from its comment list.

We define the relevance as follows: when a comment is related to persons, places, events or things appearing in the blog post, this comment is considered to be highly related to the blog post. According to this principle, we define 1 to 5 five grades to measure the relevance degree of each comment with its blog post.

First, we use CCD and CCE to segment blog post and all of their comments into individual words.

Then we calculate the similarity between each pair of words. The similarity between comment and blog content can be derived from the word similarity with the word frequency. The formula for the similarity calculation between comment and content is defined as follows:

$$Sim1(cont, comn) = \frac{\prod_{word_i, word_j} sim(word_i, word_j)}{n_1 * n_2} / MaxSim, \quad (6)$$

⁶ <http://blog.sina.com/>

⁷ <http://hi.baidu.com/>

⁸ <http://blog.sohu.com/>

⁹ <http://blog.ifeng.com/>

where $word_i \in$ word set of blog content, $word_j \in$ word set of comment. $sim(word_i, word_j)$ is the semantic similarity between $word_i$ and $word_j$ that is calculated using CCD and CCE as the semantic database. MaxSim is the maximum value of the sim values between the each pair of words of which one is from the blog content set and the other is from the comment set.

In consideration of the short comment, frequency information may be biased. So we propose another formula to calculate the similarity.

$$Sim2(cont, comn) = \frac{\sum_{word_i, word_j} \log(P_{word_j} * sim(word_i, word_j))}{n} / MaxSim, \quad (7)$$

where n is the amount of word set of comments, P_{word_j} is the probability of $word_j$ in the content.

We invite 4 volunteers to give correlation grades (1 to 5) for each comment. Finally, we define Ecc_1 and Ecc_2 , to represent the eccentricity of Sim1 and Sim2 on the base of the given scores.

$$Ecc_1 = \frac{1}{n} \sum_i \left(\frac{sim1_i}{sim1_{average}} - \frac{GivenScore_i}{GivenScore_{average}} \right)^2, \quad (8)$$

$$Ecc_2 = \frac{1}{n} \sum_i \left(\frac{sim2_i}{sim2_{average}} - \frac{GivenScore_i}{GivenScore_{average}} \right)^2. \quad (9)$$

The results are shown in Table 4. Values in CCD1 and CCE1 are calculated by formula 6, and those in CCD2 and CCE2 are calculated by formula 7. CCD1 and CCD2 are calculated with CCD as the lexical database, while CCE1 and CCE2 are calculated with CCE.

Table 4. Eccentricity of Analysis

Blog Website	CCD1	CCE1	CCD2	CCE2
Sina	0.5710	0.4926	0.4905	0.3411
Sohu	0.2724	0.2286	0.2213	0.1942
Ifeng	0.7080	0.4950	0.5492	0.4113
Baidu	0.4950	0.5996	0.3865	0.4651

The results show that in most cases, the analysis results calculated with CCE is closer to the results given by human being. For numbers of blog posts in Baidu Blog, the comments are more like a sort of chatting, so the relevance between the comments and the blog posts is weak.

5 Conclusion and Future Work

In this paper, we propose a method for ontology integration. We construct a new Chinese ontology, Chinese Concept Encyclopedia (CCE). CCE keeps the well-designed structure of an expert-edited ontology, meanwhile borrows abundant

information from an online cyclopedia. Compared to an expert-edited ontology, CCE can keep up with the emergence of new words automatically. We use the category information as clues to figure out the hypernyms of the new words to be added. From the results of some case study, we can see that the new words are added reasonably most of the time. As a semantic repository, CCE can be used in many text mining and document analysis tasks. Our experimental results show that CCE can get competitive or even better performances than CCD due to a wide coverage of new words.

Acknowledgement. We would like to thank Institute of Computational Linguistics of Peking University for permitting us to use Chinese Concept Dictionary. This work is partially supported by NSFC with Grant No. 61073081, HGJ with Grant No. 2011ZX01042-001-001, and National Key Technology R&D Pillar Program in the 11th Five-year Plan of China with Research No. 2009BAH47B05.

References

1. Dong, Z., Dong, Q.: Introduction to HowNet - Chinese Message Structure Base (2000), <http://www.keenage.com>
2. Jiang, S., Bing, L., Sun, B., Zhang, Y., Lam, W.: Ontology Enhancement and Concept Granularity Learning: Keeping Yourself Current and Adaptive. In: Proceedings of The 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, San Diego, CA, US, pp. 1244–1252 (2011)
3. Yu, J., Yu, S., Liu, Y., Zhang, H.: Introduction to Chineses Concept Dictionary. In: Proceedings of the International Conference on Chinese Computing (ICCC 2001), pp. 361–367 (2001)
4. Knight, K., Luk, S.K.: Building a Large-Scale Knowledge Base for Machine Translation. In: AAAI 1994 Proceedings (1994)
5. Kozareva, Z., Hovy, E.: A Semi-Supervised Method to Learn and Construct Taxonomies using the web. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 1110–1118 (2010)
6. Lee, S., Huh, S.-Y., McNeil, R.D.: Automatic generation of concept hierarchies using WordNet. *Expert Systems with Applications* 35(3), 1132–1144 (2008)
7. Mendes, S., Chaves, R.P.: Enriching WordNet with Qualia Information. In: Proceedings of the Workshop on WordNets and Other Lexical Resources at NAACL 2001 Conference, Pittsburgh, pp. 108–112 (2001)
8. Suchanek, F.M., Kasnecia, G., Weikuma, G.: YAGO: A Large Ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(3), 203–217 (2008)
9. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: Proceedings of the Second International Conference on Information and Knowledge Management (CIKM 1993), pp. 67–74 (1993)
10. Wolf, E., Gurevyc, I.: Aligning Sense Inventories in Wikipedia and WordNet. In: First Workshop on Automated Knowledge Base Construction, pp. 24–28 (2010)

Cluster Ensembles via Weighted Graph Regularized Nonnegative Matrix Factorization

Liang Du^{1,2}, Xuan Li^{1,2}, and Yi-Dong Shen¹

¹ State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

² Graduate University, Chinese Academy of Sciences, Beijing 100049, China
{duliang,lixuan,ydshen}@ios.ac.cn

Abstract. Cluster ensembles aim to generate a stable and robust consensus clustering by combining multiple different clustering results of a dataset. Multiple clusterings can be represented either by multiple co-association pairwise relations or cluster based features. Traditional clustering ensemble algorithms learn the consensus clustering using either of the two representations, but not both. In this paper, we propose to integrate the two representations in a unified framework by means of weighted graph regularized nonnegative matrix factorization. Such integration makes the two representations complementary to each other and thus outperforms both of them in clustering accuracy and stability. Extensive experimental results on a number of datasets further demonstrate this.

Keywords: Cluster Ensembles, Weighted Graph Regularization, NMF.

1 Introduction

It is well recognized that different clustering methods may produce different clustering results on a given data set. The reason for this is that each clustering algorithm has its own bias resulting from different criteria. Therefore, cluster ensembles have emerged in recent years as a technique to overcome such problems [1,2,3,4]. This technique is also known as clustering ensemble [5], clustering aggregation [6] or consensus clustering [7]. It reconciles multiple clustering results of a data set into a single consolidated clustering using a consensus function.

Cluster ensembles contain two key components, i.e. a representation of input clusterings and a consensus function. Two representations are currently widely used; one is *multiple co-association matrices* [2,4] and the other *cluster based features* [1]. To the best of our knowledge, current cluster ensembles approaches learn their consensus functions using either of the two representations, but not both. On the one hand, the multiple co-association matrices contain pairwise relationships of different clusterings and the averaged co-association matrix can be seen as the consensus similarity of input clusterings. On the other hand, a dataset can be represented using cluster based features, which contain connections between data points and clusters. Although both the co-association matrices and

cluster based features can be used independently to obtain consensus clustering, algorithms that make use of them simultaneously should be able to generate more meaningful clustering structures; similar idea has also been explored in clustering problem [8].

Besides, each partition may have different clustering performance. It is also required to automatically identify better clusterings from all partitions. One natural idea is to assign different weights to input clusterings based on their contributions for the cluster ensembles [24].

Based on the above observations, in this paper we propose a *Weighted Graph regularized Nonnegative Matrix Factorization* (WGNMF) model for cluster ensembles by integrating the above two representations into a unified framework and learn the weights of different clusterings. The basic idea is as follows: We construct two matrices; one is a *consensus affinity matrix* from multiple co-association matrices and the other is a *cluster feature matrix* from cluster based features. These two matrices are used in a regularization process, where we learn an implicit consensus function from the cluster feature matrix by nonnegative matrix factorization (NMF) while the factorization procedure is regularized with the consensus affinity matrix. In our WGNMF model, the weights of input clusterings are learned during the factorization process. We empirically evaluate WGNMF model over benchmark data sets, which demonstrates that it outperforms existing approaches using only one of the above two representations.

The rest of paper is organized as follows. Section 2 reviews related work on cluster ensembles. Section 3 presents our WGNMF model. Section 4 reports the experimental results, and Section 5 concludes the paper with some future work.

2 Related Work

We briefly review some related work on cluster ensembles. Previous work on learning consensus functions are mainly based on the two representations of input clusterings: the multiple co-association matrices and the cluster based features.

One popular strategy of building consensus functions is to utilize multiple co-association matrices [9] (also known as connectivity matrices in [7,24]). The averaged co-association matrix coming from input clusterings can be seen as a new data matrix in a new feature space or a similarity matrix, and thus traditional approaches operated on data matrix or similarity can be deployed. Hadjitodorov et al. [10] showed that good clustering results can be obtained by running Kmeans on the averaged co-association matrix. Li et al. used nonnegative matrix factorization (NMF) on the averaged co-association matrix to generate the final consensus clustering. Li and Ding [2] proposed to learn the weighted co-association matrix within an NMF procedure. Wang et al. proposed generalized weighted cluster aggregation (GWCA) [4] to learn the consensus function by minimizing the sum of the Bregman divergence between the consensus function and all of the co-association matrices.

Another different strategy is to construct consensus functions implicitly from the representation of cluster based features. These implicit consensus functions

can be roughly grouped in two categories. The first category is graph based approaches in which a graph is constructed from the cluster based features and some graph partition algorithms are used for the final consensus clustering. Strehl et.al. [1] proposed three graph-based approaches: Cluster-based Similarity Partition Algorithm (CSPA), HyperGraph Partition Algorithm (HGPA), and MetaClustering Algorithm (MCLA). In CSPA, A binary similarity matrix is constructed and specific algorithms like METIS [11] is used to partition the graph. HGPA utilizes the HMETIS [11] algorithm to partition the hypergraph where each hyperedge represents a cluster of an input clusterings. MCLA collapses related hyperedges and assigns each object to the collapsed hyperedge in which it participates most strongly. Fern and Brodley [12] proposed the hybrid bipartite graph partition algorithm, which partitions the bipartite graph using spectral graph partition algorithms. Al-Razgan and Domeniconi [9] proposed an approach to partitioning a weighted similarity graph. The second category makes use of probabilistic graphical models. Topchy et al. [13] proposed a probabilistic model using a finite mixture of multinomial distributions in the space of input clusterings. Wang et al. [3] proposed a generative probabilistic model Bayesian cluster ensembles (BCE) for cluster ensembles, which is derived from Latent Dirichlet Allocation (LDA) [14].

3 WGNMF

We first introduce some notation and briefly review NMF [15] which is used to learn the consensus function implicitly from cluster based representation. We then describe our WGNMF model which integrates multiple co-association matrices and the representation of cluster based features within the process of NMF. We also present specific optimization techniques for WGNMF.

3.1 Notation

Given a data set of n points and a collection of m clustering solutions $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^m\}$. Each clustering solution \mathcal{P}^c for $c = 1, \dots, m$ is a partition of the data set. A partitioning of these n points into k clusters can be represented as a set of k clusters $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ or a label vector $\mathcal{P}^c \in \mathbb{R}^n$, where k is the number of clusters. Note that the number of clusters k in different \mathcal{P}^c could be different.

The cluster-based representation [1] can be constructed as follows. For each clustering \mathcal{P}^c , we construct the binary membership indicator matrix $\mathbf{H}^c \in \mathcal{R}^{n \times k}$, where each column corresponds a cluster and each row is a point. $H_{ij}^c = 1$ if the point i is assigned to cluster j in partition c , and $H_{ij}^c = 0$ otherwise. The concatenated matrix $\mathbf{X} = (\mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^m)$ is used to represent the data matrix in a new feature space.

The co-association matrix [7] of partition \mathcal{P}^c is defined as a $n \times n$ squared matrix \mathbf{W}^c , where $\mathbf{W}_{ij}^c = 1$ if points i and j are assigned to the same cluster, and $\mathbf{W}_{ij}^c = 0$ otherwise.

Example 1. Suppose we have 5 samples and 2 clusterings each with 3 clusters. Then we have 2 partitions $\mathcal{P}^1 = [1, 1, 2, 3, 3]$ and $\mathcal{P}^2 = [2, 3, 3, 1, 1]$. The cluster-based representation \mathbf{X} and the co-association matrices \mathbf{W}^1 for \mathcal{P}^1 and \mathbf{W}^2 for \mathcal{P}^2 of these samples are given below.

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{W}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \mathbf{W}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

3.2 NMF and Extensions

Non-negative Matrix Factorization [15] is a factorization algorithm focused on the analysis of nonnegative matrix. Given a nonnegative matrix $\mathbf{X} \in \mathcal{R}_+^{n \times d}$, where each row of \mathbf{X} is a data point, NMF approximates \mathbf{X} using two low-rank nonnegative matrices $\mathbf{U} \in \mathcal{R}_+^{n \times k}$ and $\mathbf{V} \in \mathcal{R}_+^{d \times k}$. There are two cost functions commonly used to measure the quality of the approximation. The first one is the square of the Euclidean distance between two matrices

$$O_1 = \|\mathbf{X} - \mathbf{UV}^T\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ is Frobenius norm. The second one is the divergence between two matrices

$$O_2 = D(\mathbf{X} \|\mathbf{UV}^T) = \sum_{i=1}^n \sum_{j=1}^d (\mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\mathbf{Y}_{ij}} - \mathbf{X}_{ij} + \mathbf{Y}_{ij}) \tag{2}$$

where $\mathbf{Y}_{ij} = \mathbf{U}_i \mathbf{V}_j^T$.

Recently, Cai et al. [16] proposed Graph regularized NMF (GNMF). It aims at learn a compact representation which uncovers the hidden semantics and simultaneously preserves the intrinsic local geometric structure. The basic assumption behind is that if two data points are similar, then their low dimensional representations are also close to each other.

3.3 Weighted Graph Regularized NMF

In subsection 3.1, we obtained a matrix $\mathbf{X} = (\mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^m)$ from the cluster based representation of input clusterings. Here, we use NMF to find two low-rank matrices \mathbf{U} and \mathbf{V} to approximate it. Due to the close connection between NMF and clustering, the obtained \mathbf{U} can be used to extract the final consensus clustering.

To obtain a better approximation of \mathbf{X} by \mathbf{U} and \mathbf{V} , inspired by [16], we incorporate the multiple co-association matrices $\{\mathbf{W}^c\}_{c=1}^m$ into NMF. To apply this idea in cluster ensembles, we define a consensus graph, where the affinity matrix $\hat{\mathbf{W}}$ is constructed by linearly combining the given multiple co-association matrices

$$\hat{\mathbf{W}} = \sum_c^m \alpha_c \mathbf{W}^c \quad \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0$$

where $\alpha_c, c = 1, 2, \dots, m$, is the weight associated with partition \mathcal{P}^c . Recall that the co-association matrix of \mathbf{W}^c for \mathcal{P}^c can be seen as a binary similarity matrix, and it defines a coarse affinity graph. We expect that the weighed combination of multiple co-association matrices can better capture the similarities between data points. The learned weights also provide clues to select individual input clusterings. The reason is that an input clustering with larger weight contributes more to the consensus affinity graph and the final clustering.

Given the above definition of consensus affinity graph, we can use the following two functions to measure the smoothness of the low dimensional representation of data points:

$$\begin{aligned} \mathcal{R}_1 &= \frac{1}{2} \sum_{c=1}^m \sum_{i,j} \alpha_c \mathbf{W}_{ij}^c \|U_i - U_j\|^2 \\ &= \sum_{c=1}^m \alpha_c \left(\sum_i D_{ii}^c U_i U_i^T - \sum_{i,j} \mathbf{W}_{ij}^c U_i U_j^T \right) \\ &= \sum_{c=1}^m \alpha_c (\text{tr}(U^T D^c U) - \text{tr}(U^T \mathbf{W}^c U)) \\ &= \text{tr}(U^T (\sum_{c=1}^m \alpha_c L^c) U) \end{aligned} \quad (3)$$

and

$$\begin{aligned} \mathcal{R}_2 &= \frac{1}{2} \sum_{c=1}^m \sum_{i,j} \alpha_c \mathbf{W}_{ij}^c (D(U_i \| U_j) + D(U_j \| U_i)) \\ &= \frac{1}{2} \sum_{c=1}^m \sum_{i,j} \sum_l \alpha_c \mathbf{W}_{ij}^c (U_{il} \log \frac{U_{il}}{U_{jl}} + U_{jl} \log \frac{U_{jl}}{U_{il}}) \end{aligned} \quad (4)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix and \mathbf{D} is a diagonal matrix with $D_{ii}^c = \sum_j \mathbf{W}_{ij}^c$. $\mathbf{L}^c = \mathbf{D}^c - \mathbf{W}^c$ is called the graph Laplacian.

Combining the weighted affinity graph together with the above smoothness functions with NMF leads to our Weighted Graph regularized Non-negative Factorization (WGNMF) model: If the Euclidean distances is used, WGNMF minimizes the following objective function:

$$\begin{aligned} \mathcal{O}_1 &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \lambda \text{tr}(U^T (\sum_{c=1}^m \alpha_c L^c) U) \\ \text{s.t. } & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \\ & \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0 \end{aligned} \quad (5)$$

If the divergence is used, WGNMF minimizes the following objective function:

$$\begin{aligned}
\mathcal{O}_2 = & \sum_{i=1}^n \sum_{j=1}^d (\mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\sum_{l=1}^k \mathbf{U}_{il} \mathbf{V}_{jl}} - \mathbf{X}_{ij} + \sum_{l=1}^k \mathbf{U}_{il} \mathbf{V}_{jl}) \\
& + \frac{\lambda}{2} \sum_{c=1}^m \sum_{i=1}^n \sum_{j=1}^d \sum_{l=1}^k \alpha_c \mathbf{W}_{ij}^c (\mathbf{U}_{il} \log \frac{\mathbf{U}_{il}}{\mathbf{U}_{jl}} + \mathbf{U}_{jl} \log \frac{\mathbf{U}_{jl}}{\mathbf{U}_{il}}) \\
\text{s.t. } & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \\
& \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0
\end{aligned} \tag{6}$$

where the regularization parameter $\lambda \geq 0$ controls the smoothness of the low dimensional representations.

3.4 Optimization

In \mathcal{O}_1 (Eq. (5)) and \mathcal{O}_2 (Eq. (6)), there are 3 unknown variables, \mathbf{U} , \mathbf{V} and α . When two of them are fixed, the subproblem of computing the optimal value for the other variable is easy to solve. Hence, \mathcal{O}_1 and \mathcal{O}_2 can be solved by iteratively updating \mathbf{U} , \mathbf{V} and α , so that the value of the objective functions gradually decrease. This process can be viewed as a "block coordinate descent" process [17]. We describe this process for \mathcal{O}_1 and \mathcal{O}_2 separately.

Minimizing \mathcal{O}_1 . When α is fixed, the optimization problem of computing \mathbf{U} , \mathbf{V} becomes

$$\begin{aligned}
\mathcal{L} = & \text{tr}(\mathbf{X}\mathbf{X}^T) - 2\text{tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) \\
& + \lambda \sum_{c=1}^m \alpha_c \text{tr}(\mathbf{U}^T \mathbf{L}^c \mathbf{U}) + \text{tr}(\Phi \mathbf{U}^T) + \text{tr}(\Psi \mathbf{V}^T)
\end{aligned} \tag{7}$$

where Φ and Ψ are the lagrange multiplier for the nonnegative constraints.

Notice that the above optimization problem is equivalent to GNMF [16] with a combined Laplacian matrix. Thus, the estimation for \mathbf{U} and \mathbf{V} will be exactly same as that in GNMF with a combined Laplacian matrix. It is [16]:

$$\mathbf{U}_{il} = \mathbf{U}_{il} \frac{(\mathbf{X}\mathbf{V} + \lambda(\sum_{c=1}^m \alpha_c \mathbf{W}^c)\mathbf{U})_{il}}{(\mathbf{U}\mathbf{V}^T\mathbf{V} + \lambda(\sum_{c=1}^m \alpha_c \mathbf{D}^c)\mathbf{U})_{il}} \tag{8}$$

$$\mathbf{V}_{il} = \mathbf{V}_{il} \frac{(\mathbf{X}^T\mathbf{U})_{il}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{il}} \tag{9}$$

When \mathbf{U} and \mathbf{V} are fixed, the optimization problem for α is equivalent to solving the following problem

$$\min_{\alpha} \sum_{c=1}^m \alpha_c \text{tr}(\mathbf{U}^T \mathbf{L}^c \mathbf{U}), \text{ s.t. } \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0 \tag{10}$$

which is a *linear programming* problem and can be efficiently solved. However, the solution will always be

$$\alpha_c = \begin{cases} 1, & \text{if } c = \arg \min_c \text{tr}(\mathbf{U}^T \mathbf{L}^c \mathbf{U}) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

To avoid this trivial solution, we propose to add one regularization term to (10) and solve the following problem

$$\min_{\alpha} \sum_{c=1}^m \alpha_c \text{tr}(\mathbf{U}^T \mathbf{L}^c \mathbf{U}) + \lambda_2 \|\alpha\|^2, \text{ s.t. } \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0 \quad (12)$$

where $\lambda_2 \geq 0$ is a tradeoff parameter. In this way, we will solve a *quadratic programming* (QP) problem with respect to α when \mathbf{U} and \mathbf{V} are fixed.

Minimizing \mathcal{O}_2 . When α is fixed, the optimization problem of computing \mathbf{U} , \mathbf{V} becomes

$$\begin{aligned} \mathcal{L} = & \sum_{i=1}^n \sum_{j=1}^d (\mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\sum_{l=1}^k \mathbf{U}_{il} \mathbf{V}_{jl}} - \mathbf{X}_{ij} + \sum_{l=1}^k \mathbf{U}_{il} \mathbf{V}_{jl}) \\ & + \frac{\lambda}{2} \sum_{c=1}^m \sum_{i=1}^n \sum_{j=1}^d \sum_{l=1}^k \alpha_c \mathbf{W}_{ij}^c (\mathbf{U}_{il} \log \frac{\mathbf{U}_{il}}{\mathbf{U}_{jl}} + \mathbf{U}_{jl} \log \frac{\mathbf{U}_{jl}}{\mathbf{U}_{il}}) \\ & + \text{tr}(\Phi \mathbf{U}^T) + \text{tr}(\Psi \mathbf{V}^T) \end{aligned} \quad (13)$$

Similarly, the above optimization is equivalent to GNMF-KL [16] with a combined Laplacian matrix. We give the linear system for update \mathbf{U}

$$\left(\sum_j v_{jl} I + \lambda \left(\sum_c L^c \right) \right) \mathbf{u}_l = \begin{bmatrix} u_{1l} \sum_j (x_{1j} v_{jl} / \sum_l u_{1l} v_{jl}) \\ u_{2l} \sum_j (x_{2j} v_{jl} / \sum_l u_{2l} v_{jl}) \\ \dots \\ u_{nl} \sum_j (x_{nj} v_{jl} / \sum_l u_{nl} v_{jl}) \end{bmatrix} \quad (14)$$

The above linear system can be solved either by the matrix inverse or some efficient iterative algorithms. The correspond update rule for \mathbf{V} is given below.

$$\mathbf{V}_{jl} = \mathbf{V}_{jl} \frac{\sum_i (\mathbf{X}_{ij} \mathbf{U}_{il} / \sum_l \mathbf{U}_{il} \mathbf{V}_{jl})}{\sum_i \mathbf{U}_{il}} \quad (15)$$

When \mathbf{U} and \mathbf{V} are fixed, a similar QP problem with respect to the weights α becomes

$$\begin{aligned} \min_{\alpha} & \sum_{c=1}^m \alpha_c \left[\sum_{i=1}^n \sum_{j=1}^d \sum_{l=1}^k \mathbf{W}_{ij}^c (\mathbf{U}_{il} \log \frac{\mathbf{U}_{il}}{\mathbf{U}_{jl}} + \mathbf{U}_{jl} \log \frac{\mathbf{U}_{jl}}{\mathbf{U}_{il}}) \right] + \lambda_2 \|\alpha\|^2 \\ \text{s.t.} & \sum_{c=1}^m \alpha_c = 1, \alpha_c \geq 0 \end{aligned} \quad (16)$$

Algorithm 1. Minimizing \mathcal{O}_1 and \mathcal{O}_2

Input: Partitions $\{\mathcal{P}^i\}_{i=1}^m$, the number of clusters k and regularization parameter λ **Output:** U , V and α

- 1: Construct m co-association matrix $\{W^c\}_{c=1}^m$
 - 2: Initialize U , V and set $\alpha = [1/m, 1/m, \dots, 1/m]^T$
 - 3: **repeat**
 - 4: Compute the weighted graph Laplacian based on α
 - 5: Update U_{il} according to Eq. (8) for \mathcal{O}_1 or (14) for \mathcal{O}_2
 - 6: Update V_{jl} according to Eq. (9) for \mathcal{O}_1 or (15) for \mathcal{O}_2
 - 7: Update α by solving the QP programming in Eq. (12) for \mathcal{O}_1 or (16) for \mathcal{O}_2
 - 8: **until** convergence
 - 9: **return** U , V and α
-

The whole process of estimating the low-rank matrices U , V and the combination weights α is described in Algorithm 1.

The above learning algorithm will converge. In each iteration the objective function value always decreases. On the one hand, when U and V are fixed, WGNMF solves a quadratic programming problem. On the other hand, when α is fixed WGNMF boils down to GNMF, the update rules of U and V are similar to the update rules in GNMF and therefore Cai's [16] proof can also be applied.

4 Experiments

In this section, we conduct experiments on a number of real-world datasets to evaluate of the effectiveness of the proposed method.

4.1 Datasets

We use a variety of datasets (see Table 1) to evaluate the accuracy of the proposed method. The number of classes is ranged from 3 to 40, the number of samples ranged from 47 to 2340, and the number of dimension ranged from 4 to 21839. Details are as follows:

- Five datasets (Iris, Glass, Ecoli, Soybean, Zoo) are from UCI data repository [18].
- The COIL dataset is an image library from Columbia with 20 objects. The images of each objects were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images.
- The ORL dataset is a face database from Olivetti Research Laboratory. It consists of 400 face images with 40 people (10 samples per person). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses).
- The WebACE dataset is from WebACE project and has been used for document clustering. It contains 2340 documents consisting of news article from Reuters new service of 20 different categories collected in October 1997.

Table 1. Descriptions of datasets.

Data sets	Samples	Dimensions	Classes
Iris	150	4	3
Glass	214	9	6
Ecoli	336	7	8
Soybean	47	35	4
Zoo	101	18	7
COIL	1440	1024	20
ORL	400	1024	40
WebACE	2340	21839	20
Tr11	414	6429	9
Tr12	313	5804	8

- Tr11, Tr12, these datasets are derived from the TREC [\[1\]](http://trec.nist.gov) collection, which are often used in document clustering.

The size of each image in COIL and ORL is 32×32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector.

4.2 Evaluation Criteria

In the experiments, we set the number of clusters equal to the number of classes k for all the cluster ensembles algorithms. To evaluate their performance, we compare the clustering results generated by these algorithms with the true classes by computing two performance measures, Clustering Accuracy (Acc) and Normalized Mutual Information (NMI) [\[1\]](#).

For a sample x_i , the cluster label assigned by the algorithm is denoted as r_i , and ground true label is y_i . The accuracy is defined as follows:

$$\text{Acc} = \frac{\sum_{i=1}^n \delta(y_i, \text{map}(r_i))}{n}$$

where n is the total number of samples and $\delta(x, y)$ is the indicator function that equals 1 if $x = y$ and equals 0 otherwise, and $\text{map}(r_i)$ is the permutation mapping function that maps the obtained labels r_i to the equivalent label from the data set. The best mapping function can be found by using the Kuhn-Munkres algorithm [\[19\]](#). The value of Acc equals 1 if and only if the clustering result and the true label are identical. Larger values of Acc indicate better clustering performance.

Given a clustering \mathcal{P} and the true partitioning \mathcal{Y} (class labels). The number of clusters in \mathcal{P} and classes in \mathcal{Y} are both k . Suppose n_i is the number of objects in the i -th cluster, n_j is the number of objects in the j -th class and n_{ij} is the number of objects which belongs to the i -th cluster and j -th class. NMI between \mathcal{P} and \mathcal{Y} is calculated as follows [\[1\]](#):

¹ <http://trec.nist.gov>

$$\text{NMI}(\mathcal{P}, \mathcal{Y}) = \frac{\sum_{i=1}^k \sum_{j=1}^k n_{ij} \log \frac{n \cdot n_{ij}}{n_i \cdot n_j}}{\sqrt{\sum_{i=1}^k n_i \log \frac{n_i}{n} \sum_{j=1}^k n_j \log \frac{n_j}{n}}}$$

The value of NMI equals 1 if and only if \mathcal{P} and \mathcal{Y} are identical and is close to 0 if \mathcal{P} is a random partitioning. Larger values of NMI indicate better clustering performance.

Table 2. Experimental Results in Clustering Accuracy

	Kmeans	KC	CSPA	HGPA	MCLA	BCE	GWCA	WGNMF-Euc	WGNMF-KL
Iris	0.81	0.85	0.87	0.62	0.89	0.89	0.89	0.89	0.89
Glass	0.42	0.49	0.43	0.40	0.46	0.49	0.53	0.54	0.51
Ecoli	0.65	0.64	0.56	0.51	0.61	0.66	0.64	0.67	0.65
Soybean	0.72	0.65	0.69	0.72	0.73	0.70	0.73	0.75	0.73
Zoo	0.69	0.67	0.58	0.55	0.74	0.67	0.74	0.77	0.73
COIL	0.59	0.62	0.69	0.55	0.69	0.67	0.58	0.69	0.71
ORL	0.50	0.53	0.58	0.60	0.60	0.51	0.52	0.56	0.60
WebACE	0.43	0.46	0.40	0.35	0.47	0.48	0.47	0.46	0.48
Tr11	0.52	0.57	0.49	0.47	0.52	0.58	0.58	0.60	0.60
Tr12	0.47	0.56	0.54	0.52	0.57	0.58	0.58	0.57	0.60

4.3 Comparison Settings

To generate the input for cluster ensembles algorithms, we adopt a common strategy [3] by running Kmeans 200 times with random initiation and then splitting each 20 clustering results as input. In this way, we repeat the cluster ensembles algorithms 10 times. We report the averaged performance over 10 rounds.

To demonstrate how the clustering performance can be improved by our method, we compare the following clustering ensemble algorithms.

- Standard Kmeans clustering algorithm (Kmeans).
- Kmeans clustering on Consensus matrix (KC). The consensus function is defined as the averaged co-association matrix.
- The Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and MetaClustering Algorithm (MCLA) are three algorithms described in [1]. We use the author’s matlab implementation ClusterPack².
- Bayesian Cluster Ensembles [3] (BCE) is a generative probabilistic model which learns the implicit consensus function from the cluster-based representation.
- Generalized Weighted Cluster Aggregation [4] (GWCA) define the consensus function by the weighted averaged co-association matrix. We use the Euclidean distance to learn the weighted consensus matrix and the spectral clustering algorithm³ is further used to derive the final clustering.

² www.lans.ece.utexas.edu/~strehl

³ <http://www.cis.upenn.edu/~jshi/software/>

Table 3. Experimental Results in Normalized Mutual Information

	Kmeans	KC	CSPA	HGPA	MCLA	BCE	GWCA	WGNMF-Euc	WGNMF-KL
Iris	0.69	0.72	0.71	0.39	0.74	0.74	0.75	0.74	0.74
Glass	0.31	0.33	0.29	0.26	0.32	0.35	0.37	0.38	0.37
Ecoli	0.58	0.59	0.51	0.40	0.56	0.59	0.57	0.59	0.58
Soybean	0.72	0.67	0.63	0.69	0.71	0.69	0.71	0.73	0.71
Zoo	0.69	0.70	0.59	0.60	0.74	0.70	0.73	0.74	0.73
COIL	0.73	0.75	0.76	0.69	0.78	0.77	0.73	0.79	0.80
ORL	0.71	0.74	0.76	0.77	0.77	0.69	0.73	0.75	78
WebACE	0.53	0.55	0.51	0.45	0.54	0.57	0.56	0.58	0.57
Tr11	0.48	0.58	0.52	0.48	0.52	0.60	0.56	0.59	0.60
Tr12	0.38	0.54	0.49	0.43	0.50	0.57	0.50	0.57	0.59

We present the results of our WGNMF algorithm under the Euclidean distance and KL divergence (denoted as WGNMF-Euc and WGNMF-KL). The regularization parameter λ is selected via a coarse grid search process.

4.4 Experimental Results

The experimental results are summarized in Tables 2 and 3. From these two tables we observe that our WGNMF improves Kmeans clustering on all datasets. Moreover, WGNMF (WGNMF-Euc or WGNMF-KL) achieves the best performance on 9 out of 10 datasets and its performance on the remaining datasets is close to the best results. In summary, the experiments clearly show the effectiveness of weighted graph regularized NMF for improving the cluster ensemble algorithms in terms of clustering accuracy and NMI.

4.5 Individual Clustering Selection

A useful byproduct of WGNMF is that the learned weights can be used to select individual input clusterings, i.e. assess how important each input clustering is. The reason is that an input clustering with a larger weight contributes more to the consensus affinity graph and the final clustering. We compare the top-5 selected clusterings based on the weights learned from WGNMF and GWCA [4], the latter is a weighted consensus clustering technique and also learns the combination weights, with the results of all clusterings. The results are given in Figure 1. We observe that the input clusterings which obtain larger weights are generally good clusterings.

4.6 Impact of Parameter λ

In our model λ is used to control the smoothness of the low-dimensional representations. We run WGNMF with varying λ from a coarse grid (0.1, 1, 10, 100). Table 4 shows the impact of λ on Acc and NMI with some datasets. We observe that WGNMF achieves good performance in a wide range and it is easy to tune.

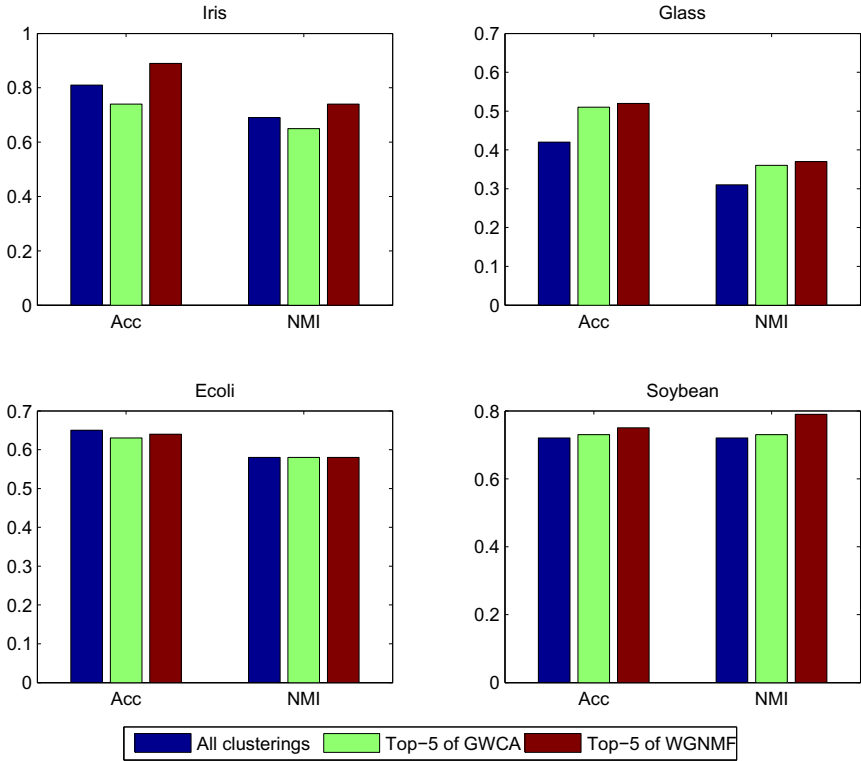


Fig. 1. Performance comparison of All clusterings vs. Top-5 clusterings

Table 4. The performance of WGNMF vs. parameter λ

	Acc				NMI			
	0.1	1	10	100	0.1	1	10	100
Iris	0.89	0.89	0.89	0.89	0.74	0.74	0.74	0.74
Glass	0.53	0.54	0.52	0.52	0.37	0.38	0.37	0.37
Soybean	0.73	0.73	0.73	0.75	0.71	0.71	0.71	0.73
Zoo	0.75	0.75	0.77	0.68	0.73	0.74	0.74	0.70

5 Conclusion and Future Work

In this paper, we propose a weighted graph regularized nonnegative matrix factorization model for cluster ensembles task. We integrate two representations of input clustering, multiple co-association matrices and cluster feature matrix in a unified regularization framework. We learn the implicit consensus function from cluster feature matrix by NMF procedure while the factorization is regularized with consensus matrix. The weights of input clusterings are learned within the factorization process. Extensive experiments on a number of real-world datasets

demonstrate that the proposed method mostly outperforms the other cluster ensemble algorithms.

In future work, we want to extend the idea of this paper to other statistical models, such as topic modeling, to integrate different representations of input clusterings beyond the multiple co-association matrices and the cluster based matrix.

Acknowledgments. We would like to thank all anonymous reviewers for their helpful comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) grants 60970045 and 60833001.

References

1. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research* 3, 583–617 (2002)
2. Li, T., Ding, C.: Weighted consensus clustering. In: *Proceedings of the 8th SIAM International Conference on Data Mining*, pp. 798–809 (2008)
3. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. In: *Proceedings of the 9th SIAM International Conference on Data Mining*, pp. 211–222 (2009)
4. Wang, F., Wang, X., Li, T.: Generalized cluster aggregation. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1279–1284 (2009)
5. Topchy, A., Jain, A., Punch, W.: Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1866–1881 (2005)
6. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data* 1(1), 4 (2007)
7. Li, T., Ding, C., Jordan, M.: Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: *Proceedings of the 7th IEEE International Conference on Data Mining*, pp. 577–582 (2007)
8. Wang, F., Ding, C., Li, T.: Integrated kl (k-means-laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In: *Proceedings of the 9th SIAM International Conference on Data Mining*, pp. 38–48 (2009)
9. Al-Razgan, M., Domeniconi, C.: Weighted clustering ensembles. In: *Proceedings of 6th SIAM International Conference on Data Mining*, pp. 258–269 (2006)
10. Hadjitodorov, S., Kuncheva, L., Todorova, L.: Moderate diversity for better cluster ensembles. *Information Fusion* 7(3), 264–275 (2006)
11. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 20(1), 359 (1999)
12. Fern, X., Brodley, C.: Solving cluster ensemble problems by bipartite graph partitioning. In: *Proceedings of the 21th International Conference on Machine Learning*, pp. 281–288 (2004)
13. Topchy, A., Jain, A., Punch, W.: A mixture model for clustering ensembles. In: *Proceedings of 4th SIAM International Conference on Data Mining*, pp. 379–390 (2004)
14. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022 (2003)

15. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755), 788–791 (1999)
16. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (to appear, 2011)
17. Bertsekas, D.: *Nonlinear programming*. Athena Scientific, Belmont (1999)
18. Asuncion, A., Newman, D.J.: *UCI machine learning repository* (2007)
19. Lovász, L., Plummer, M.: *Matching theory* (1986)
20. Fern, X., Brodley, C.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: *Proceedings of the 20th International Conference on Machine Learning*, pp. 186–193 (2003)
21. Munkres, J.: Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 32–38 (1957)
22. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* 52(1), 91–118 (2003)

Continuously Identifying Representatives Out of Massive Streams^{*}

Qiong Li^{1,2}, Xiuli Ma^{1,2,**}, Shiwei Tang^{1,2}, and Shuiyuan Xie^{1,2}

¹ School of Electronics Engineering and Computer Science, Peking University,
Beijing, China, 100871

² Key Laboratory on Machine Perception (Ministry of Education), Peking University,
Beijing, China, 100871

{liqiong,maxl}@cis.pku.edu.cn, tsw@pku.edu.cn

Abstract. More and more emerging applications are involved in monitoring multiple data streams concurrently. In these applications, the data flow out of multiple concurrent sources continuously. In such large-scale real-time monitoring applications, continuously identifying representatives out of massive streams is an important task which aims to capture key trends to support online monitoring and analysis. In this paper, we present a framework for continuously extracting representatives out of massive streams. Our framework identifies and traces representatives based on core clustering technique. We adapt the core clustering model under streaming condition and propose a method of extracting representatives by utilizing the advantage characteristic of core clusters that core set is tight. In order to continuously identify the representatives in an efficient way, we apply online representatives adjust processes only when significant clustering evolution happens. As shown in our experimental studies, our algorithm is effective and efficient.

Keywords: data mining, data streams, representative, core clustering.

1 Introduction

More and more emerging applications are involved in monitoring multiple data streams concurrently. Such applications include network monitoring, sensor networks, financial data analysis and so on. In such large-scale real-time monitoring applications, the data flow out of multiple concurrent sources continuously. In many cases, streams vary at different numerical levels but with similar trends or shapes. For example, in large distributed network of drinking water, the chlorine concentration levels of some nodes may rise and fall coherently over time. It is significant to reduce the scale of monitoring by utilizing the correlation among streams. As a result, conti-

^{*} This work is supported by the National Natural Science Foundation of China under Grant No.61103025.

^{**} Corresponding author.

Continuously identifying representatives out of massive streams is an important task which aims at capturing key trends to support online monitoring and analysis.

Traditional methods on monitoring and analysis applications have mainly focused on a single data stream [1]. However, in large-scale monitoring applications, it is not the raw sensory readings associated with a single node that is interesting. Identifying representatives out of massive streams in real-time will give a clear overview of the key trends of data across the entire system. Pattern discovery in multiple streams is another kind of work which focuses on discovering patterns in order to summarize streams [2] [10]. Most of these methods use some mathematical transformation tools to discover and model the resulted patterns which can summarize and compress the original streams efficiently. But these methods lack an intuitive interface to represent and explain the result to end users. It is crucial to continuously mining some more interpretable and intuitive results. Clustering over multiple streams [3] [5] [6] is an effective way to summarize multiple streams that puts the similar streams together and separates dissimilar ones apart. Clusters can provide information of similarities between streams. However, clustering cannot directly present the key trends out of massive streams. As a result, we extract representatives beyond performing clustering technique on streams. And we use correlation as our similarity measure because it is common that two different streams vary with similar trends or shapes in large-scale real-time monitoring applications. In this paper, we present a framework named *Continuously Identifying Representatives out of massive Streams* abbreviated as CIRS. Explicitly, this framework will identify and trace several representatives based on correlation clustering technique.

Traditional clustering models have two main shortcomings. One shortcoming is that the number of clusters needs to be predefined. Another shortcoming is the clusters are not tight, which means some pairs of objects in one cluster may have small similarities. We hope that every pair of streams in one cluster has high similarity. As a result, we consider the model of pair-wise clustering [7] [16] which does not need to predefine the number of clusters and can guarantee that the similarity of any pair of objects in one cluster is high enough. But the complexity of common pair-wise clustering algorithms is insufferable. So we adapt core clustering model [8] which have a complexity of polynomial to complete our clustering process. However, the original core clustering model is designed for high dimensional data. It is not suitable for streaming conditions. Thus, we expand the model for streams. The core clustering model partitions data set into some core sets and boundary sets. The similarity of any pair of objects in one core set is not less than a given threshold. This characteristic is called that the core set is tight. The similarity of any pair of objects from different core sets is less than this threshold. Thus the dissimilar streams can be well separated. Because of introducing local optimization, the complexity of the core clustering algorithm is polynomial which is quite less than that of common pair-wise clustering.

Continuously identifying representatives faces several challenges: (1) Massive streams update at a very fast rate. For a large-scale real-time monitoring application with tens of thousands to millions of nodes, identifying representatives effectively becomes computationally challenging. Clustering can assist with solving this problem, because the clustering results can provide information of similarities between

streams. So how to identify representatives based on a sound clustering model is very important. (2) Near real-time monitoring and analysis of incoming data streams is required. For example, in drinking water monitoring system, outbreak detection requires immediate response and cannot afford any offline-processing. So representatives should be identified in real time. The evolution of clusters can support the extraction of representatives continuously and efficiently. So we must propose an effective method for clustering over streams in an online way. Checking clustering evolutions periodically [5] may cause a waste of time if the clusters remain almost the same. On the other hand, the evolvments information of clusters may be missed due to the fast changing rate of data. We will propose an effective method for clustering streams dynamically.

Consequently, we first execute core clustering on the streams. Then we extract representatives from the clusters. Finally, we adjust the representatives timely when clusters evolution happens, so that we can identify the representatives out of massive streams continuously. The main contributions of our work are listed as follows:

- We propose a method of identifying representatives out of massive streams effectively. The method is based on core clustering and utilizes the characteristic of core clusters that the core set is tight. Thereby the representatives can be identified with high qualities.
- We adapt the core clustering model under streaming condition. The new model will favor the online adjustments of clusters and continuously identifying representatives.
- We propose a framework of continuously adjusting representatives efficiently. The framework focuses on the clusters evolvments. The representatives will be adjusted as long as significant clustering evolution happens.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3, the problem statement is given. The main strategy of identifying representatives is described in Section 4. Section 5 presents the experimental results. Finally, this paper concludes with Section 6.

2 Related Work

Pattern discovery in multiple streams or time series continues to attract high interest [2] [10]. SPIRIT [2] summarizes key trends by calculating hidden variables incrementally. Dynammo [10] focuses on summarization of coevolving sequences with missing values. Most of these methods use some mathematical transformation tools such as PCA to discover and model the resulted patterns. The results are not intuitive enough to represent and explain to end users. It is crucial to identify and track representatives in a more interpretable and intuitive way.

Clustering over streams is another common method to summarize and analyze massive data streams. Various research works have been reported to deal with this problem. These works can be divided into two main kinds. One kind of works focuses

on clustering the data points [11] [12] [13]. They regard the streams as distributed sources of the original data. They cluster the data points not the streams. Another kind of works discusses clustering the streams as our point of view. A COD framework is provided in [3]. It summarizes data online and clusters offline when users give queries. The method proposed in [5] utilizes a periodical way of checking cluster evolutions. The cluster split or merge processes will be performed when cluster evolves. The periodical method can cause a waste of time if the clusters remain almost the same, while the cluster information may be lost if the data change in a fast rate. COMET-CORE [6] is a framework of continuously clustering streams by events. It defines events as the marked changes of the streams. It computes correlation between approximated streams and continuously clusters them by cluster splitting and merging. However, COMET-CORE adopts traditional clustering model. The clusters are not tight. In addition, COMET-CORE will not consider the streams which cannot be assigned into any present clusters. This may miss some important information. Clustering results cannot intuitively present the key trends which are essential for reducing the scale of monitoring in large-scale online monitoring and analysis.

In conclusion, none of the proposed frameworks can achieve the goal of continuously identifying representatives out of massive streams with efficient and intuitive form.

3 Problem Statement

First, we denote a stream in the form of $S [1, 2, \dots, t]$, where t is the latest timestamp of the stream. Then, we can define a set of streams as $\{S_1, S_2, \dots, S_n\}$ for ease of presentation.

We use correlation as our similarity measure because it is common that two different streams vary with similar trends or shapes in large-scale real-time monitoring applications. We adopt Pearson correlation coefficient in this paper. $S_i [k]$ denotes the value at timestamp k of the data stream S_i . So we give the definition of correlation between two streams as follow.

Definition 3.1. (CORRELATION COEFFICIENT) *the correlation coefficient between two streams S_i and S_j is*

$$corr(S_i, S_j) = \frac{\sum_k S_i[k] \cdot S_j[k] - \sum_k S_i[k] \sum_k S_j[k]}{\sqrt{\sum_k (S_i[k] - \bar{S}_i)^2} \sqrt{\sum_k (S_j[k] - \bar{S}_j)^2}}$$

On this basis, we can give the definition of representative.

Definition 3.2. (REPRESENTATIVE) *Given a threshold δ_c , if the correlation coefficient between $S_i \in \{S_1, S_2, \dots, S_n\}$ and $S_j \in \{S_1, S_2, \dots, S_n\}$ satisfies $corr(S_i, S_j) \geq \delta_c$, then S_i can represent S_j and S_i is a representative of S_j .*

This definition describes the representative in a mathematical way. We can infer from Definition 3.2 that the definition is symmetric. If S_i is a representative of S_j , then S_j is a representative of S_i . And we can draw from this further that any stream is a representative of itself. However, we hope to continuously identify some representatives that can represent as many streams as possible because these representatives are more helpful for capturing the key trends of all streams. So we define an indicator to evaluate the quality of a representative.

Definition 3.3. (COVERAGE RATE) *Given a stream $S_r \in \{S_1, S_2, \dots, S_n\}$, n_r is the number of streams which can be represented by stream S_r in the set $\{S_1, S_2, \dots, S_n\}$. Then the coverage rate of stream S_r is n_r/n . Given a streams set $R = \{S_{r1}, S_{r2}, \dots, S_{rm}\} \subseteq \{S_1, S_2, \dots, S_n\}$, n_R is the number of streams which can be represented by at least one stream in R . Then the coverage rate of set R is n_R/n .*

Coverage rate indicates the effectiveness of a representative. The higher the value is, the more streams can be represented. We hope to continuously identify a smallest set of representatives which have a high coverage rate.

Finally, we state our problem as follow: given a coverage rate $c\%$ and a correlation threshold δ_c , how to find the smallest set of representatives $R = \{S_{r1}, S_{r2}, \dots, S_{rm}\}$ at every timestamp, which can represent at least $c\%$ streams of the streams set $\{S_1, S_2, \dots, S_n\}$.

4 Continuously Identifying Representatives

At the beginning of this section, the brief concept of the framework is given. Due to the shortcomings of existing clustering models, we adopt one of the pair-wise clustering models called core clustering [8]. The result of core clustering can ensure that the similarity between any pair of objects in one core set is not less than a given threshold. So it can support the extraction of representatives effectively. In order to identify representatives continuously, we have to update the result of core clustering timely. Instead of re-clustering all streams, we adjust cluster results online when there are significant changes so that the representatives can be extracted effectively.

4.1 Core Clustering

Original core clustering model [8] is designed for high-dimensional data which are common in some applications such as microarray in gene engineering. Unlike high-dimensional data, streams cannot be stored in advance. So the processing on streams cannot be completed offline. The resulted patterns on streams should be updated whenever new data arrive. As a result, we adapted the core clustering model to meet

the online requirement. We proposed real-time adjustment strategies under streaming condition in order that the clustering results can be updated as long as new data arrive. Furthermore, we used correlation as our similarity measure because it is common that two different streams vary with similar trends or shapes in large-scale real-time monitoring applications.

Definition 4.1. (CORE SET) C is a core set of the streams set $\{S_1, S_2, \dots, S_n\}$ if $\text{corr}(S_i, S_j) \geq \delta_c$ for $\forall S_i, S_j \in C$, where $\text{corr}(S_i, S_j)$ is the correlation coefficient between S_i and S_j , δ_c is a given correlation threshold.

The correlation coefficient between any two streams in the same core set is not less than a given threshold. We denote this characteristic as that the core set is tight. This characteristic will advantage the identifying of representatives, because all the streams in one core set can be represented by only one stream.

Core sets can constitute clusters directly. However, this will lead to many small clusters. We do not hope this happen, because small clusters do not favor the extraction of representatives. Therefore boundary set is proposed. We give the definition of boundary set for stream condition as follow.

Definition 4.2. (BOUNDARY SET) B is the boundary set of core set C , where $B = \{S \mid \text{corr}(S, S_i) \geq \delta_c, S_i \in C\}$.

Boundary set is the set of streams which meet the challenge of correlation threshold δ_c with at least one stream in the corresponding core set. A cluster will be composed of a core set and a corresponding boundary set, denoted as a pair like (C, B) . The goal of core clustering is to partition the streams set into several core sets having no intersection and their corresponding boundary sets.

Based on these definitions, we can constitute some initial clusters as a foundation of online adjustments. The algorithm for constituting initial clusters has two steps. First step is to calculate the correlation matrix of all streams in the streams set, which stores the correlation coefficient between any pair of streams. We adopt the method proposed in [9] to calculate the correlation coefficients efficiently. The method applies a sliding window on streams. It takes the Discrete Fourier Transform and adopts weighted Euclidean distance as the correlation coefficient. The details of the algorithm are passed over here. The second step is the process of partitioning the streams into several core sets and their corresponding boundary sets. Core clustering model adopts a local optimization policy [8]. We also adapt this under streaming condition. First, we get the stream S which satisfies the correlation threshold with the maximum number of streams in the current streams set. Then we generate the core set containing S and the corresponding boundary set. The stream which has the highest correlation with S and satisfies the correlation threshold with all the streams in the current core set will be added into the core set every time. And the boundary set will accordingly be updated. If the stream satisfying the above restrict does not exist, the core set and the corresponding boundary set are generated completely. Then remove these streams from the current streams set and get back to the start to generate a new core set and

corresponding boundary set. The algorithm will finish if the current streams set becomes empty. This strategy can reduce the complexity of the algorithm to polynomial.

Thus, we can get the core clustering results over the streams set $\{S_1, S_2, \dots, S_n\}$, which can be denoted as $\{(C_1, B_1), (C_2, B_2), \dots, (C_n, B_n)\}$, where (C_i, B_i) is a cluster in the form of core set and its corresponding boundary set.

4.2 Extracting Representatives

Based on the initial result of core clustering, we identify the representatives. As the core set is tight, any stream of a core set can be treated as a representative of the streams in its cluster. However, considering the purpose of representing as many streams as possible, we finally identify the stream correlative to the maximum number of streams in the corresponding boundary set as the representative of the cluster. This method directly utilizes the advantage characteristic of core clustering. The correlation coefficient between any two streams in the same core set is not less than the given correlation threshold. So the representative can represent all the streams in the same core set. In addition, the representative can also represent the maximum number of streams in the boundary set. Thus it can represent the maximum number of streams in the cluster. On this basis, we propose the definition of expanded core set, which extends the conception of core set.

Definition 4.3. (*EXPANDED CORE SET*) (C, B) is a cluster. S_r is the representative of the cluster. We call $ExpC = C \cup \{S_i \in B \mid corr(S_i, S_r) \geq \delta_c\}$ the expanded core set.

Expanded core set contains all of the streams correlative to the representative. The scale of expanded core set reflects the quality of representative directly. The more streams expanded core set contains, the higher the coverage rate of the representative is, and the higher quality the representative has. Algorithm 1 shows the main steps of extracting representatives.

Algorithm 1. ExtractRS

Input: the streams set, the correlation threshold δ_c .

Output: the clusters and the representatives.

1 Partition the streams set into several core sets and corresponding boundary sets by core clustering;

2 **for** every cluster

3 check streams in core set C ;

4 identify the stream S correlative to the most streams in the boundary set B as the representative;

5 create expanded core set $ExpC$ for current cluster;

6 **end for**

7 **return** the clusters and the representatives;

We can get initial representatives so far. Next we will focus on how to adjust them in real time due to clusters evolvments while data arrive continuously.

4.3 Representatives Adjustment

The correlation between streams will change very likely when new data arrive. And clusters will evolve accordingly. As a result, the representatives must be updated as long as the clusters evolve. How to detect cluster evolvments in time will be the key point. As mentioned in Section 2, the periodical way may cause a waste of time if the clusters remain almost the same. Or the evolvments information may be lost due to the fast changing rate of clusters. Another way is defining some events to denote the evolving of clusters, like COMET-CORE [6]. But COMET-CORE adopts a traditional clustering model, which cannot support the extraction of representatives effectively. And the results of COMET-CORE always include a temporary set, in which streams will not be considered. In this paper, we detect the changes of clusters by observing the cluster members. If members in one cluster cannot be represented by the representative anymore, we will adjust the clusters by transferring these members. The transferring of members focuses on maintaining the core sets tight while reducing the processing time as far as possible, so that the quality of core clusters can be ensured while the adjustments also meet the real time challenge. The particular steps are listed as follows. t_c is the current timestamp.

Algorithm 2. AdjustRS

Input: the present clustering results, the present representatives, the correlation threshold δ_c , re-clustering threshold δ_r .

Output: the clusters and the representatives after adjustments.

```

1 if there exists a cluster that the percentage of streams cannot be represented
  is large than  $\delta_r$ 
2   re-cluster;
3 end if
4 for every cluster
5   check the expanded core set  $ExpC$ ;
6   if  $S_c$  in  $ExpC$  cannot be represented by representative stream
7     delete  $S_c$  from  $ExpC$ ;
8     if there is another representative can represent  $S_c$ 
9       add  $S_c$  into the corresponding boundary set of the most correlative
representative ;
10    else
11       $S_c$  composes a new cluster;
12    end if
13  end if
14 end for
15 for every cluster

```

```

16  check the boundary set  $B$ ;
17  if  $S_b$  in  $B-ExpC$  can be represented by representative stream
    //  $B-ExpC$  is the set of members in  $B$  but not in  $ExpC$ .
18    add  $S_b$  into  $ExpC$ ;
19  end if
20 end for
21 update the representatives of all adjusted cluster;

```

It needs two steps to complete the member transferring,

Check expanded core set of every cluster, if the stream S_c in a certain expanded core set which no longer meets the correlation threshold with the representative, then S_c will be removed from this expanded core set. If there exists other representatives which can represent S_c , then assign S_c into the expanded core set of the most similar representative accordingly. Or else if there is a stream which is strongly correlative with S_c , then assign S_c into the boundary set of the corresponding cluster of this stream. Otherwise, S_c will compose a new cluster on its own.

Check boundary set of every cluster, if stream S_b in a certain boundary set and not in the corresponding expanded core set meets the challenge of the correlation threshold with the representative of the current cluster, S_b will be adjusted into the expanded core set.

The two steps mentioned above will complete the member transferring due to clusters evolvments. And we will achieve the goal of adjusting clusters. At last, we re-extract representatives of adjusted clusters. The main steps are listed in Algorithm 2.

Notice that we first check every cluster. If there exists a cluster that the streams are no longer correlative with the representative exceed the given threshold, we will re-cluster all the streams. This step can avoid low efficiency due to too many member transferring caused by the large amount of streams which cannot be represented any more. The complete description of CIRS is listed in Algorithm 3.

Algorithm 3. CIRS

Input: the streams set, the correlation threshold δ_c , re-clustering threshold δ_r .

Output: the clusters and the representatives of every timestamp.

```

1  get initial clusters and representatives by calling ExtractRS;
2  if new data come in
3    AdjustRS;
4    update the clusters and the representatives;
5  end if

```

Until now we have described the complete framework for continuously identifying representatives out of massive streams. At last, we get back to the problem stated in

Section 3: given a coverage rate $c\%$, we will obtain a set of representatives by the method above, which can represent at least $c\%$ of all streams and the set of representatives is smallest. We adopt an intuitive method considering of the size of the expanded core sets. We sort the clusters by the size of their expanded core set, and add the representatives into the final result set in order until the current set of representatives can meet the requirement of coverage rate.

In large-scale real-time monitoring applications, users are concerned with the typical trends of data streams. The representatives fitly meet the demand. The more streams a representative can represent, the more typical it is, and the more important it is for users. In conclusion, our method considers the typicality of representatives, and always returns the top-k representatives to users, where k is related with a given coverage requirement.

5 Experimental Evaluations

In this section, we evaluate the proposed method on monitoring data streams of real application. Section 5.1 introduces the datasets. Section 5.2 studies the effectiveness. Scalability of our method is discussed in Section 5.3. We mainly compare our method with COMET-CORE proposed in [6]. COMET-CORE is an algorithm for clustering streams. According to the clustering method, we consider the streams which have marked changes as representatives, because COMET-CORE will split and merge clusters according to the similarities between these streams and other. Thus we can experiment on the same criteria. We conducted all the following experiments on Windows XP Professional operating system equipped with an Intel Pentium 1.86GHz processor and 1 GB of RAM.

5.1 Datasets

We use the monitoring data streams of water distributed networks. Drinking water quality monitoring is a typical large-scale real-time monitoring application. We used two distribution networks of different scales. One water distribution network is a real water distribution system referred to in BWSN (The Battle of the Water Sensor Network) [15]. This network is comprised of 129 nodes. Another network is from the Centre for Water Systems at the University of Exeter [4]. It is comprised of 920 nodes.

The dataset of water distribution network was generated by EPANET 2.0 [14] that accurately simulates the hydraulic and chemical phenomena within drinking water distribution systems. And in our experiments, to construct datasets of different scales, we monitor the chlorine concentration level on two networks for different size of nodes and timestamps. We will simply denote the dataset as the form of *nodes*timestamps*. For example dataset of 129 nodes and 648 timestamps will be denoted as 129*648.

5.2 Effectiveness

In this section, the quality of representative is discussed. We probe two criteria: the percentage of representatives in all streams and the average correlation between representatives and the streams they represent. The coverage rate is 100% and the correlation threshold is 0.8 for both COMET-CORE and our method. The experiments were conducted on different size of nodes and timestamps. Fig.1 shows the results versus timestamps and Fig.2 shows the results versus nodes. As shown in Fig.1, our method CIRS is more effective than COMET-CORE. In Fig. 1(a), the percentage of representatives of CIRS is lower than COMET-CORE at most timestamps. CIRS can use just less than 5% streams to represent other streams at best, while COMET-CORE will use more than 10% streams. And CIRS can use about 40% streams to represent all streams at worst, while COMET-CORE uses more than 70% streams. As shown in Fig. 1(b), average correlation between representatives and the streams they represent of our method is obviously higher than that of COMET-CORE. The average correlation of our method is always close to 1, while the result of COMET-CORE is about 0.9. This profit from the characteristic of core cluster that every core set is tight. And our adjusting strategies can also ensure the quality of the clusters. In conclusion, our method can continuously identify representatives in a very effective way. As shown in Fig.2, our method is still more effective than COMET-CORE. In Fig. 2(a), the

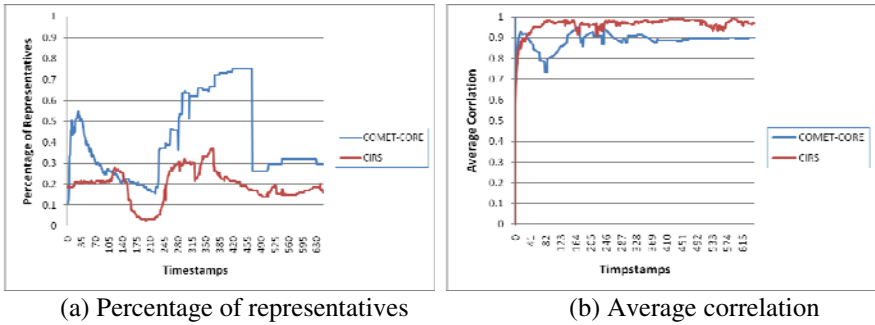


Fig. 1. Effectiveness vs. timestamps

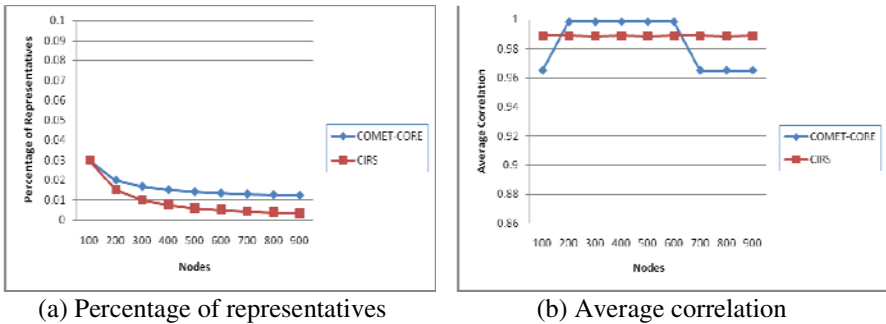


Fig. 2. Effectiveness vs. nodes

percentage of representatives of our method is lower than COMET-CORE in each case. And the result becomes better when the size of nodes is larger. In Fig. 2(b), the average correlation of our method is obviously higher than that of COMET-CORE in general though the results in some cases are a little lower than COMET-CORE. In conclusion, our method is more effective than COMET-CORE for different size of data.

5.3 Efficiency

We compare our method with two other methods: one is called Basic here, and the other is COMET-CORE. The Basic algorithm also extracts representatives based on clustering. It does clustering periodically and identifies the centers of clusters as representatives. We repeated every experiment 10 times and calculated the average runtime as our final results. The results on different datasets are shown in Fig. 3 and Fig.4. Fig.3 shows the average runtime on datasets of different nodes size. Our method CIRS is much more efficient than Basic in all cases. Compared with COMET-CORE, our method is a little slower. The main reason of this is that our clustering model is a pair-wise clustering model which considers the similarity between every pair of streams.

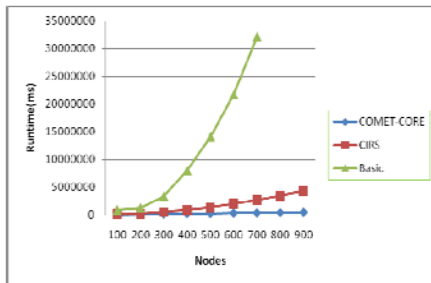
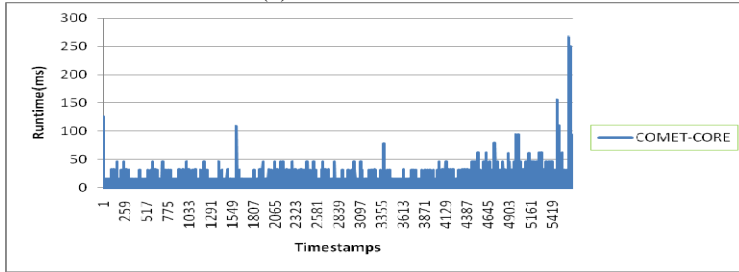


Fig. 3. Runtime vs. nodes

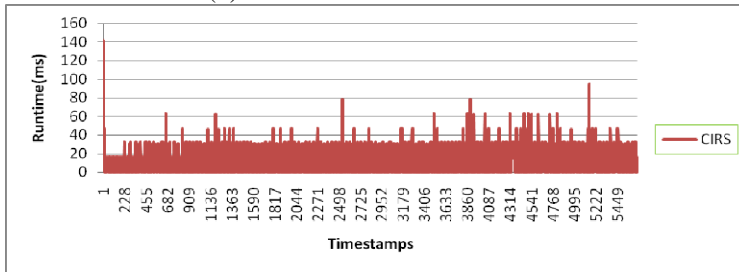
The runtime versus timestamps is presented in Fig. 4. The result of Basic is shown in Fig. 4(a). Though the runtime almost remains the same at different timestamps, the result is much more inefficient than other two methods. The processing time of Basic on streams which continuously have new data arrive will become intolerable. COMET-CORE and CIRS are much better than Basic. The results are shown in Fig. 4(b) and Fig. 4(c) respectively. The average runtimes of two methods are about the same. But we can confirm from Fig. 4(d) that the runtime of our method remains stable when the size of timestamps becomes larger while the result of COMET-CORE has a trend to become larger and larger. As a result, the processing time of CIRS on streams of different timestamps is linear with the size of timestamps while the result of COMET-CORE seems to be exponential with it. Consequently, our method will be much more efficient when streams come continuously.



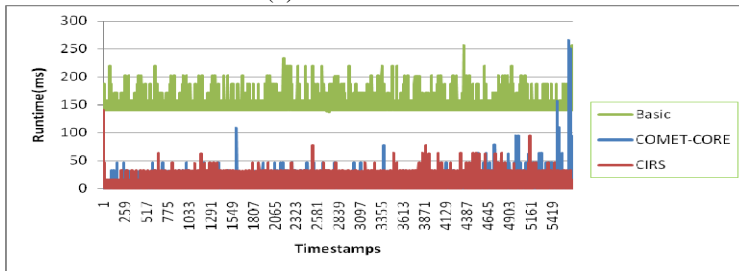
(a) Runtime of Basic



(b) Runtime of COMET-CORE



(c) Runtime of CIRS



(d) Runtime of three methods

Fig. 4. Runtime vs. timestamps

6 Conclusion

In this paper, we proposed a framework for continuously identifying representatives out of massive streams. Clustering over multiple streams is an effective way to

summarize multiple streams. However, clustering cannot directly present the key trends out of massive streams. As a result, we extract representatives based on performing clustering technique on streams. Considering the shortcomings of traditional clustering models, we finally adopt a pair-wise clustering model which is called core clustering. This model is designed for high dimensional data, so we adapt it under streaming condition and propose a method of identifying representatives by utilizing the characteristic of core clusters. We also apply efficient representatives adjusting processes only when significant clusters evolution happens. As validated in the experimental results, our algorithm is effective and efficient.

Moreover, we look forward to discover the changing patterns of representatives for events detection in large-scale real-time applications.

References

1. CANARY User's Manual, VERSION 4.2.,
<http://www.epa.gov/NHSRC/news/news122007.html>
2. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming pattern discovery in multiple time-series. In: VLDB (2005)
3. Dai, B.-R., Huang, J.-W., Yeh, M.-Y., Chen, M.-S.: Adaptive Clustering for Multiple Evolving Streams. *IEEE Trans. Knowledge and Data Eng.* 18(9), 1166–1180 (2006)
4. Center for Water System at University of Exeter, <http://centres.exeter.ac.uk/cws>
5. Rodrigues, P.P., Gama, J., Pedroso, J.P.: ODAC: Hierarchical Clustering of Time Series Data Streams. In: Proc. Sixth SIAM Int'l Conf. Data Mining, pp. 499–503 (2006)
6. Yeh, M., Dai, B., Chen, M.: Clustering over Multiple Evolving Streams by Events and Correlations. *TKDE* 19(10), 1349–1362 (2007)
7. Wang, H., Wang, W., Yang, J., et al.: Clustering by Pattern Similarity in Large Data Sets. In: The Int'l Conf on Management of Data, Madison (2002)
8. Jiang, L., Yang, D., Tang, S., Ma, X., Zhang, D.: A Core Clustering Approach for Cube Slice. *Journal of Computer Research and Development*, 359–365 (2006)
9. Mueen, A., Nath, S., Liu, J.: Fast approximate correlation for massive time-series data. In: SIGMOD (2010)
10. Li, L., McCann, J., Pollard, N., Faloutsos, C.: DynaMMO: Mining and Summarization of Coevolving Sequences with missing values. In: SIGKDD (2009)
11. Zhou, A., Cao, F., Yan, Y., Sha, C., He, X.: Distributed Data Stream Clustering: A Fast EM-based Approach. In: ICDE (2007)
12. Cormode, G., Muthukrishnan, S., Zhuang, W.: Conquering the Divide: Continuous Clustering of Distributed Data Streams. In: ICDE (2007)
13. Zhang, Q., Liu, J., Wang, W.: Approximate Clustering on Distributed Data Streams. In: ICDE (2008)
14. Rossman, L.A.: EPANET2 user's manual. National Risk Management Research Laboratory: U.S. Environmental Protection Agency (2000)
15. Ostfeld, A., Uber, J.G., Salomons, E.: Battle of water sensor networks: A design challenge for engineers and algorithms. In: WDSA (2006)
16. Jiang, L., Yang, D., Tang, S., Ma, X., Zhang, D.: Mining Maximal Correlated Member Clusters in High Dimensional Database. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 149–159. Springer, Heidelberg (2006)

Cost-Sensitive Decision Tree for Uncertain Data^{*}

Mingjian Liu¹, Yang Zhang^{1, **}, Xing Zhang¹, and Yong Wang²

¹ College of Information Engineering, Northwest A&F University, China
zhangyang@nwsuaf.edu.cn

² School of Computer, Northwestern Poly technical University, China

Abstract. Uncertainty exists widely in real-world applications. Recently, the research for uncertain data has attracted more and more attention. While not enough attention has been paid to the research of cost-sensitive algorithm on uncertain data. In this paper, we propose a simple but effective method to extend traditional cost-sensitive decision tree to uncertain data, and the algorithm can deal with both certain and uncertain data. In our experiment, we compare the proposed algorithm with DTU[18] on UCI datasets. The experimental result proves that the proposed algorithm performs better than DTU, with lower computational complexity. It keeps low cost even at high level of uncertainty, which makes it applicable to real-life applications for data uncertainty.

Keywords: Cost-sensitive, Uncertain Data, Decision Tree.

1 Introduction

In the research field of classification analysis, accuracy is the traditional criterion for measuring classification model, such as C4.5[16], DTU[18] and UDT[19]. Their goal is to minimize the misclassification error. However, these models are valid only when different types of classification errors have the same cost, which doesn't hold in real-life applications. For example, in medical diagnosis, the cost for misdiagnosing a patient to be healthy may be far greater than that misdiagnosing a healthy person as being sick, because the former may lead to the loss of life. Meanwhile, in order to maintain high accuracy, these models do as many tests as possible, which neglect the enormous cost tests may generate, such as some medical tests are very expensive. Thus researchers propose the cost-sensitive learning with the objective to minimize the expected total cost of tests and misclassifications.

Recently, there are amounts of works on cost-sensitive researching. To the best of our knowledge, these researches only can handle certain data. However, data-uncertainty exists widely in real-world data, such as medical diagnosis, sensor network, market analysis, etc. Many reasons contribute to the emergence of uncertainty, such as imprecise

^{*} This work is supported by the National Natural Science Foundation of China (60873196) and Chinese Universities Scientific Fund (QN2009092).

^{**} Corresponding author.

measurement, outdated sources, decision errors, etc. Uncertainty can be numerical, such as the data collection of sensor network for temperature and humidity. And uncertainty categorical data can appear in medical diagnosis, such as doctor have difficulty to accurately judge a tumor to be benign or malignant due to precision limitation of experiment. So judgment will be supported with certain probability or confidence [20].

Since data uncertainty is ubiquitous, it is important to develop research for uncertain data. In this paper, we borrow the idea in DTU[18] and UDT[19] to handle data uncertainty, and use the probabilistic cardinality to build cost-sensitive decision tree for uncertain data.

Our main contribution is we firstly propose a Cost-Sensitive Decision Tree for Uncertain Data (CSDTU). We also be first one to integrate uncertainty data model into cost-sensitive learning. In experiment, we compare CSDTU with DTU on UCI datasets, and the certain data is treated as uncertainty 0. The result shows the effectiveness and rationality of CSDTU, which has satisfactory performance even with high uncertainty.

This paper is organized as follows. Section 2 reviews the related work. Section 3 describes the problem definition of our work. Section 4 presents the training and testing algorithm of our classification model. The experiment result and discussion is given in section 5. And section 6 concludes this paper and gives our future work.

2 Related Work

Cost-sensitive learning: cost-sensitive learning has received increasing attentions recently. In [2], Turney analyzed a whole variety of costs in machine learning, and pointed out that the most important two types of cost are the misclassification costs, that are the costs incurred by misclassification errors, and test costs that are the costs incurred for obtaining values of attribute. Some works only use misclassification costs such as [10][11]; while only attribute costs was used in [12][13]. They were pointed out to be bias in [2]. The best method should consider both misclassification and test cost. In that way, in [3], Turney designed ICET, which uses a genetic algorithm to build a decision tree to minimize the cost of tests and misclassifications. And [15] proposed a naïve Bayesian based cost-sensitive learning algorithm, called CSNB.

Many works has been done on cost-sensitive decision tree, such as [4-9], which has been accepted by public. In [4], Ling proposed a decision tree with minimum total cost of tests and misclassifications. In [6-7][9], the decision tree designed by [4] was improved with expected cost-reduction and discount in attribute criterion, and several test strategies also be proposed, which can promote the rationality and efficiency of test. In [14], Qin, etc, based on cost-sensitive tree takes the cost into source cost and target cost for researching.

Data Uncertainty: much works have been done in clustering, classification [18][19], frequent item mining and outlier detection. In [17], Aggarwal, etc, made a survey on uncertain data mining. And in [20] A rule-based algorithm on uncertain data was proposed.

DTU[18] and UDT[19] both are the decision tree to handle uncertain data. UDT builds model with probability density function and fractional tuples. And we borrow the idea of probabilistic cardinality in DTU to build classifier to handle uncertain data. So our algorithm have the same framework as DTU, and the minimal total cost of tests and misclassifications is used as the attribute split criterion. In addition, our algorithm has many properties of cost-sensitive learning, such as pointed out by [2][3]. For instance, if the misclassification cost is smaller than all test costs, then few tests will be performed, even no test, only a one-node decision tree will be returned.

3 Problem Definition

Here, we use a specific scenario to introduce the cost-sensitive classification for uncertain. In medical diagnosis, doctor will ask a patient (instance) to finish additional medical tests (attributes), which may obtain values at cost (test cost), and through the obtained value, to diagnose the disease of the patient (class). The test can improve the accuracy of diagnosis, but the error it includes will also result in uncertainty of data., that will lead to misdiagnoses. Misdiagnoses (misclassification) may also bear a cost. It contains two kinds of mistake, False-Positive (FP), which means a healthy person is misdiagnosed as sick, and False-Negative (FN), which means the patient is misdiagnosed as healthy. The cost of FN is usually larger than that of FP, as the former may incur the loss of life, while the latter leads to unnecessary medical treatment which may have bad side effects and waste money. The cost of correct diagnosis True-Positive (TP) and True-Negative (TN) are 0. For convenience, we use FP and FN to represent directly the cost of FP and FN, in the rest of paper, so are the TP and TN. We build a classification model from the training data with uncertainty.

4 Cost-Sensitive Decision Tree for Uncertain Data (CSDTU)

4.1 Data Uncertainty

If the uncertain data is numerical uncertain data, it has uncertain numerical attribute (UNA) denoted by A^{u_n} , whose value is represented as an interval and the probability distribution function(PDF) over this interval. While uncertain categorical data has uncertain categorical attributes (UCA) A^{u_c} , the attribute value v of A^{u_c} is characterized by probability distribution over categorical domain, so the value v of attribute is a probability vector $P=\langle p_1, \dots, p_n \rangle$, instead of a single attribute value for certain data.

4.2 Training Algorithm for CSDTU

In this paper, we only consider categorical attribute and binary classification, and it's easy to extend it to numerical attribute and multi-classification such as C4.5 does. In DTU[18], probabilistic cardinality[18] is proposed to calculate probabilistic entropy[18], which can calculate the uncertain data's information gain for selecting the

splitting attribute. The probabilistic cardinality [18] on categorical attribute is defined as:

The value v of A^{u_c} is represented by the probability vector $P = \langle p_1, \dots, p_n \rangle$, $P(A^{u_c} = v_k) = p_k$, and $\sum_{k=1}^n p_k = 1$ ($1 \leq k \leq n$). The probabilistic cardinality of v_k of the j th attribute $A_j^{u_c}$ is the sum of the probabilities of each instance i whose attribute value of $A_j^{u_c}$ equals to v_k :

$$PC(v_k) = \sum_{i=1} P(A_j^{u_c} = v_k) \tag{1}$$

Furthermore, the probabilistic cardinality of v_k for class C_l is:

$$PC(v_k, C_l) = \sum_{i=1} P(A_j^{u_c} = v_k \wedge C = C_l) \tag{2}$$

For certain data, we use the number of instances to calculate information entropy, while for uncertain data, the probabilistic cardinality is used to calculate probabilistic entropy.

In order to minimize the total cost, cost-sensitive decision tree algorithm [4] uses cost reduction as the splitting criterion, instead of entropy reduction in C4.5. For building cost-sensitive decision tree on uncertain data, in this paper, we propose probabilistic cost reduction(PCR) to select splitting attribute. Besides, we assume that all attributes of training instance are unknown when building, and we must pay some cost for getting the attribute value, that is different from Ling’s building algorithm. And for comparison, our algorithm is also under static cost structure, and the attribute’s cost will not be changed after it is assigned. Test cost and misclassification cost are on the same cost scale.

Probabilistic cost reduction (PCR) is defined as the difference between the total cost without splitting and the total cost after splitting on given dataset, the former is expected total misclassification cost of dataset without splitting, and the latter includes expected total misclassification cost of all subsets after splitting and the test cost of splitting attribute. After calculating PCR of each attribute, we choose the attribute with maximum PCR as splitting attribute.

Definition. Probabilistic Cost Reduction:

$$PCR = ExpMisCost(D) - (\sum_{i=1} ExpMisCost(S_i) + TestCost(A)) \tag{3}$$

Here, $ExpMisCost(D)$ denotes the expected total misclassification cost of dataset D , which will be defined latter. It gives a more accurate choice for attribute selection than only misclassification cost of one class, that is introduced in detail in [7]. S_i denotes the i -th subset of dataset D after split; and $TestCost(A)$ denotes the test cost of attribute A .

A simple example is given in Fig.1 to illustrate our idea. The dataset has P positive and N negative instances on node T . After T is split into two branches, there are $P1$ positive and $N1$ negative instances on $T1$, while $P2$ positive and $N2$ negative instances on $T2$. So the PCR can be measured by:

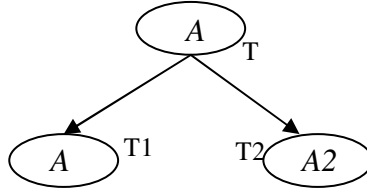


Fig. 1. A simple tree

$$PCR = ExpMisCost(T) - (ExpMisCost(T1) + ExpMisCost(T2) + TestCost(A)) \quad (4)$$

Here, $TestCost(A)$ is the test cost of splitting attribute A , which is defined as:

$$TestCost(A) = (PC(P) + PC(N)) * Cost(A) \quad (5)$$

Here, $Cost(A)$ is test cost of attribute A , $PC(P)$ and $PC(N)$ are the probabilistic cardinality of positive and negative instances on node T .

In (3), $ExpMisCost(T)$ is the expected total misclassification cost, which is firstly defined in [7]. We extend that to uncertain data as $ExpMisCost(T)$; We define MP as total misclassification cost of all positive instances on node T , $MP = PC(N) * FP$; and MN is that of all negative instances, $MN = PC(P) * FN$. Actually, choosing the attribute with maximal PRC means choosing attribute with the minimal total cost, so the attribute with the smaller misclassification cost has the larger probability to be chosen.

Consequently, the probability for positive class is: $Pr_p = 1 - \frac{MP}{MN + MP} = \frac{MN}{MN + MP}$.

So the expected misclassification cost of being positive is: $Pr_p * MP$. And the

probability for negative class is: $Pr_n = \frac{MP}{MN + MP}$. The expected misclassification

cost of being negative is: $Pr_n * MN$. In that way, $ExpMisCost(T)$ is defined as:

$$ExpectedMis(T) = Pr_p * MP + Pr_n * MN = 2 * \frac{PC(N) * FP * PC(P) * FN}{PC(P) * FN + PC(N) * FP} \quad (6)$$

With all above definitions, we calculate the PRC of attribute. $PCR > 0$ denotes that choosing the attribute can help to reduce the total cost, otherwise, the attribute can't be chosen as splitting attribute. If there are many attributes with $PCR > 0$, we choose the one with the maximal PCR.

After splitting attribute is selected, branches are grown corresponding to the each value of splitting attribute. Due to uncertainty, the weight of instance multiplies by the probability P of each attribute's value, and that be assigned to the branch corresponding to attribute's value. So one instance' weight will be assigned to many branches. When all attributes' PCR aren't greater than 0, the node is labeled as leaf. For certain data, the cost-sensitive tree labels leaf node with the class of the minimum misclassification cost, for instance, there are P positive and N negative instances on leaf node, if $P * FN > N * FP$, then the leaf is labeled as positive. In this way, the class label of leaf is

not only related to the number of instances, but also to the misclassification cost. While for uncertain data, the leaf will not be labeled definitely. The probabilistic cardinality of each class is calculated and saved on the leaf. It denotes the probability distribution of classes. The building algorithm of CSDTU is in algorithm 1. It has the same computational complexity with C4.5, and we expect it be more efficient.

Algorithm 1. Cost-Sensitive Decision Tree for Uncertain Data (CSDTU)

Input: D, Uncertain training dataset
 List of attributes' test cost
 FP, cost for false positive instance
 FN, cost for false negative instance

Output: A uncertain cost-sensitive decision tree

- 1、 Create a node T
- 2、 If (all instances are of the same class, positive (or negative))
 return root as a leaf and save (PC(P), 0) (or (0, PC(N))) on the node
- 3、 If (attributes is empty)
 return the root as a leaf and save (PC(P), PC(N)) on the node
- 4、 Calculate PRC for each attribute
- 5、 If(maximum PRC of attribute \leq 0)
 return root as a leaf and save (PC(P), PC(N)) on the node
- 6、 Else
 - a、 Let A be the attribute with maximum PRC among all the remaining attributes
 - b、 Root \leftarrow A
 - c、 For (each possible value v_i ($i=1, \dots, n$) of the attribute A)
 - i、 Grow a new branch D_i below root corresponding to $A = v_i$
 - ii、 For (each instance E_j of D)
 Put it into D_i with $E_j.\text{weight} * E_j.p(v_i)$
 - iii、 If there are no instances in the branch, add a leaf node in this branch with saving (0,0) on leaf.
 - iv、 Else call CSDTU to build a sub-tree below this branch

END
 Return root

4.3 Testing Algorithm for CSDTU

The testing process starts from the root. Due to uncertainty, test instance has the probabilistic values $\langle p_1, p_2, \dots, p_k \rangle$ corresponding to each value of splitting attribute. The weight of instance multiplies by each probabilistic values, and the results are assigned to its corresponding branch. For example, instance E 's weight multiplied by p_1 is assigned to branch 1. So test instance will be assigned into many branches. The process recursively runs on sub-node until arriving to leaf. On the leaf, we use the

probabilistic cardinality of each class to calculate the proportion of class PL , such as $PL(pos)$ denotes the probability that instance is positive. It is defined:

$$PL(pos) = \frac{PC(pos)}{PC(leaf)} \tag{7}$$

Here, $PC(leaf)$ is the probabilistic cardinality of all instances on leaf, $PC(pos)$ is the probabilistic cardinality of all positive instances on leaf.

We assume a route L from root to a leaf, and there are t tests in this route. An uncertain instance E has the probability $P(E_i)$ to reach this branch at test attribute i , so the probability for instance to be positive along L can be defined as follows:

$$P_{pos}^L = PL(pos) \times \prod_{i=1}^t P(E_i) \tag{8}$$

Because an uncertain instance can pass many routes to many leaves, and we suppose that there are m routes. All involved routes should be considered for classification analysis. So the probability that an instance is classified as positive on the whole tree can be defined as $P_{pos} = \sum_{i=1}^m P_{pos}^i$. Traditional cost sensitive decision tree use minimal misclassification cost for classification[4]. While in this paper, we use expected misclassification cost (EMC). We write EMC_{pos} and EMC_{neg} for positive and negative respectively. Taking all m routes into consideration, they can be defined as follows:

$$E. EMC_{pos} = E.MP + E.TP \tag{9}$$

$$E. EMC_{neg} = E.MN + E.TN \tag{10}$$

Here, $E.MP$ is the misclassification cost on all routes that instance E passed as positive instance, $E.MP = P_{pos} \times FP$. $E.MN$ is the misclassification cost as negative instance, $E.MN = P_{neg} \times FN$. The true cost $E.TP$ and $E.TN$ is 0;

If $EMC_{pos} < EMC_{neg}$, then instance E is labeled as positive, otherwise as negative. After classification, if the predicted label is the same as the true label of instance, the misclassification cost is 0; otherwise misclassification cost is incurred by false predicted label, such as FP is the cost incurred by false positive. Finally, the total cost for testing an instance is defined:

$$TotalCost = MisCost + TotalTC \tag{11}$$

Here, $MisCost$ is the misclassification cost, $TotalTC$ is the cost of all tested attributes, which is defined as follows:

$$TotalTC = AttrTestCost1 + AttrTestCost2 + \dots + AttrTestCostN \tag{12}$$

Here, $AttrTestCostN$ is the test cost of the N -th attribute.

5 Experiments

In this section, we report our experimental results. We study the performance of algorithm on minimizing the total cost including misclassification cost and test cost, which are the most important measurement on study of cost-sensitive tree[4]. And we compare CSDTU with DTU, these two algorithms are implemented based on WEKA.

As currently there is no real-life uncertain dataset publicly available by the research community[17-19], we need to convert existent certain data into uncertain data. We introduce synthetic uncertainty into these datasets following the method in DTU[18]. When we introduce 10% uncertainty, attribute A_i^{uc} will have the probability 0.9 to take the original value, and 0.1 to take other values. For instance, in original certain dataset, $E.A_i^{uc} = v_1$, so we assign $p_{i1} = 0.9$, and assign $p_{ik} (2 \leq k \leq n)$ to ensure $\sum_{k=2}^n p_{ik} = 0.1$. In the rest of paper, 10% uncertainty is denoted by u0.1, and the certain dataset is denoted by u0.

13 datasets are chosen from the UCI Repository for experiment, because they have binary classes and a good number of instances. The detailed information are listed in table 1. Pos/Neg represents the percentage of positive instances against negative instances. As our algorithm can only deal with categorical attribute, so we need to discrete numerical attributes with minimal entropy method[21] in advance.

Then on a PC with Core 2 CPU and 2.0GB main memory, experiment is conducted by ten-fold cross validation. To make comparison possible, we simply assign static values to the costs. Test cost of attribute is assigned by random values between \$0 and \$100; and the proportion of misclassifications' costs FP/FN is set to 600/1000,

Table 1. Datasets used in experiments

dataset	#attribute	#instance	class distribution (pos/neg)
Breast-w	10	699	458/241
Vote	17	435	267/168
Hepatitis	20	155	32/123
Car	7	1733	1211/522
Bank	11	600	274/326
Breast-cancer	10	286	201/85
Ecoli	8	336	220/116
Heart-statlog	14	270	150/120
Kr-vs-kp	37	3196	1669/1527
Tic-tac-toe	10	985	322/626
Credit-a	16	690	307/383
Diabetes	9	768	500/268
Sonar	61	208	97/111

1000/1000, and 1000/2000 respectively, so that we can study the classification performance of algorithm with different FP/FN. The uncertainty is selected from 0 to 0.5, increasing by 0.1 each time.

Fig.2 shows the average experimental results on 13 datasets with three different FP/FN settings. The curves in figure represent the average total cost of two algorithms. From that, we can see CSDTU always outperforms DTU. At u_0 , when dataset is certain, CSDTU's performance is similar to DTU, even it is higher than DTU. But with the increase of uncertainty, the cost of DTU increases apparently, while CSDTU keeps stable and even declines.

It could be observed easily that the total cost of DTU suddenly declines when uncertainty reaches 0.5. This phenomenon occurs similarly on all datasets. The reason for that can be found from table 2. In that, we choose 4 dataset who have more apparent performance to explain the reason, and their experimental results of DTU when $FP/FN=1000/1000$ is listed, including cost, accuracy and size of tree. We can see, at $u_{0.5}$, the size of tree is smaller than other uncertainties, which contributes a lot to decreasing test cost, and finally results in small total cost. And few tests may also lead

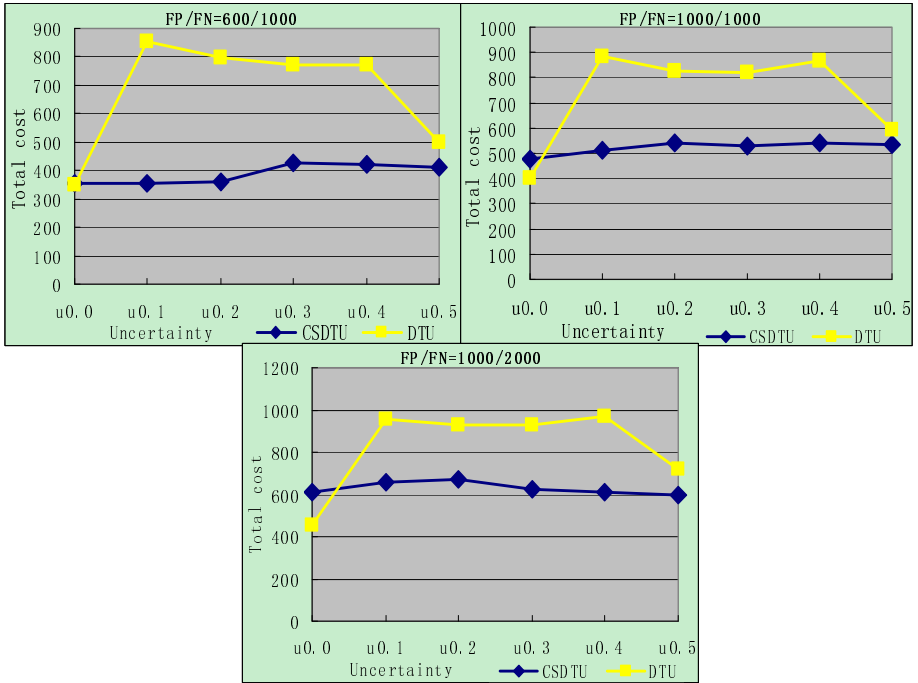


Fig. 2. Average total cost on 13 datasets

Table 2. The basic experimental result on 4 datasets

U	Hepatitis			Breast-cancer			Credit-a-disc			Vote		
	Cost	Acc	Size	Cost	Acc	Size	Cost	Acc	Size	Cost	Acc	Size
u0.0	539.69	0.5846	24	604.416	0.6716	303.1	448.751	0.7696	378.3	277.043	0.9013	31.8
u0.1	919.78	0.7875	28824	859.974	0.7128	611.4	749.158	0.8435	1441.9	840.516	0.9382	482.8
u0.2	928.11	0.7792	31955	813.914	0.7589	565.1	750.608	0.842	1212.5	842.144	0.9335	455
u0.3	986.03	0.7213	82502	823.861	0.7065	530.7	758.145	0.8333	1083.1	740.495	0.9174	469.2
u0.4	1153.9	0.5533	166127	697.633	0.703	525.5	807.45	0.7507	1105.7	969.154	0.6138	515.8
u0.5	491.5	0.6425	10.6	653.879	0.703	457.4	528.576	0.6681	900	386.205	0.6138	1

Table 3. The experimental result of CSDTU and DTU

		Tic-tac-toe	Bank	Kr-vs-kp	Vote
Average time	CSDTU	13.4	28.16667	1580.833	197
	DTU	1023.25	211.5	27423.83	151.3333
Average cost	CSDTU	363.3216	461.3981	478.09	466.486
	DTU	669.1498	715.9307	1213.72	675.9264

to the increase of misclassification cost. But it has little impact to total cost. So the phenomenon of declining suddenly at u0.5 happens. Based on this, we generate an idea to improve our algorithm in future, through studying the relevance of cost and information gain to reduce total cost by increasing accuracy.

In Table 3, we give the average total cost for prediction and average time for building on above 4 datasets when FP/FN=1000/1000. Through this simple comparison, we can illustrate CSDTU has lower cost and higher efficiency than DTU. And the similar performance happens at other different sets of FP/FN.

We also give the experimental results of these four datasets, in Fig.3, to furtherly illustrate that CSDTU is more stable and better. From that, we can see that CSDTU has the lower cost than DTU. There are similar performances on all datasets, except for the dataset *sonar*. So the experimental result of *sonar* is given in Fig.4. On that, DTU always has lower cost than CSDTU, except at u0.1. With the increase of uncertainty, CSDTU increases gradually, but DTU descends smoothly. We find CSDTU has more nodes than DTU, so the more test cost will be generated. Both of two algorithms is weak to classify the instance of *sonar*, and especially bad at high uncertainty.

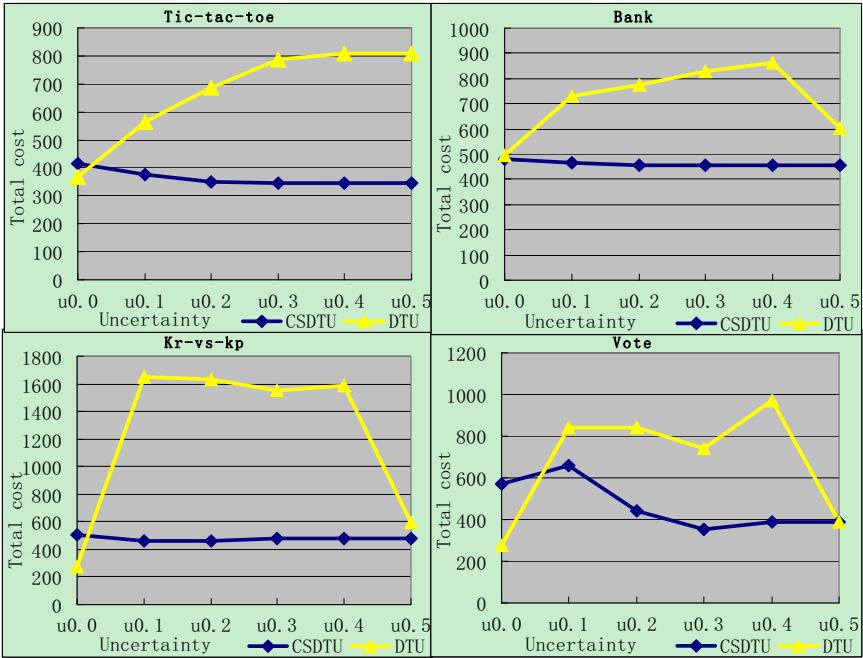


Fig. 3. Total cost on 4 datasets under FP/FN=1000/1000

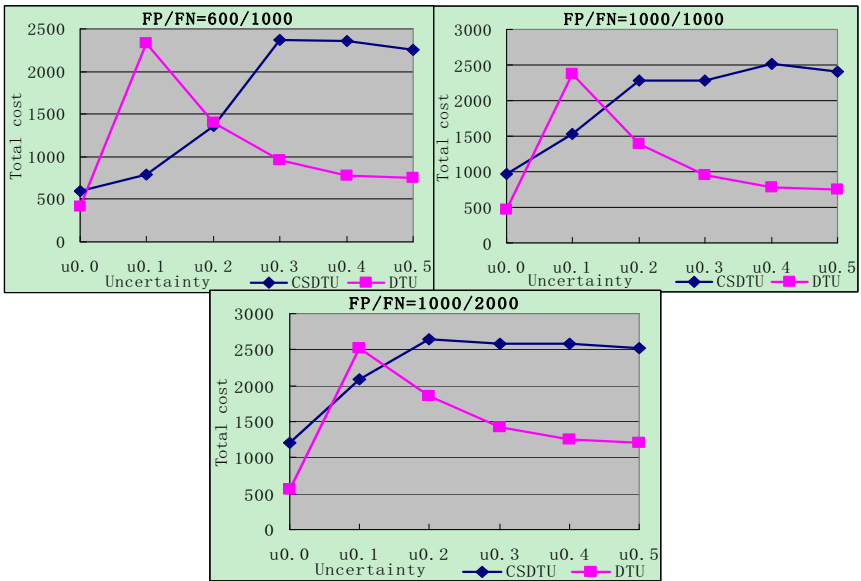


Fig. 4. The result on datasets sonar

6 Conclusions and Future Work

In this paper, we present a new algorithm CSDTU, which is a cost-sensitive decision tree for uncertain data. We improve the measures that choosing splitting attribute and testing instance on traditional cost-sensitive decision tree. Our experiments demonstrate CSDTU has a lower cost consumption and computational complexity than DTU. Even at high uncertain ratio, CSDTU still perform excellent.

In the future, we will continue to consider discounts and delayed cost in our work, even extend the methods of minimizing total cost when groups of tests must be decided together, rather than in a sequential manner.

Acknowledgments. The authors gratefully acknowledge the help from Charles X. Ling, Chen Li, and Shirui Pan.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (website). University of California, Department of Information and Computer Science, Irvine, CA (1998)
2. Turney, P.D.: Types of Cost in Inductive Concept Learning. Workshop on Cost-Sensitive Learning. In: ICML (2000)
3. Turney, P.D.: Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *JAIR* 2, 369–409 (1995)
4. Ling, C.X., Yang, Q., Wang, J., Zhang, S.: Decision Trees with Minimal Costs. In: ICML (2004)
5. Zhang, S., Qin, Z., Ling, C.X., Sheng, S.: Missing is Useful: Missing Values in Cost-sensitive Decision Trees. *IEEE Transactions on Knowledge and Data Engineering, TKDE* (2005)
6. Ling, C.X., Sheng, S., Yang, Q.: Intelligent Test Strategies for Cost-sensitive Decision Trees. *IEEE Transactions on Knowledge and Data Engineering, TKDE* (2005)
7. Sheng, S., Ling, C.X., Yang, Q.: Simple Test Strategies for Cost-Sensitive Decision Trees. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) *ECML 2005. LNCS (LNAI)*, vol. 3720, pp. 365–376. Springer, Heidelberg (2005)
8. Sheng, S., Ling, C.X.: Hybrid Cost-Sensitive Decision Tree. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) *PKDD 2005. LNCS (LNAI)*, vol. 3721, pp. 274–284. Springer, Heidelberg (2005)
9. Ling, C.X., Sheng, S., Yang, Q.: Test Strategies for Cost-Sensitive Decision Trees. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 18(8) (2006)
10. Elkan, C.: The Foundations of Cost-Sensitive Learning. In: Proceedings of the 17th International Joint Conference of Artificial Intelligence, Seattle, pp. 973–978 (2001)
11. Domingos, P.: MetaCost: A General Method for Making Classifiers Cost-Sensitive. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, pp. 155–164 (1999)
12. Greiner, R., Grove, A., Roth, D.: Learning Cost-sensitive Active Classifiers. *Artificial Intelligence* 139(2), 137–174 (2002)

13. Tan, M.: Cost-sensitive Learning of Classification Knowledge and its Applications in Robotics. *Machine Learning Journal* 13, 7–33 (1993)
14. Qin, Z., Zhang, S., Zhang, C.: Cost-Sensitive Decision Trees with Multiple Cost Scales. In: Webb, G.I., Yu, X. (eds.) *AI 2004. LNCS (LNAI)*, vol. 3339, pp. 380–390. Springer, Heidelberg (2004)
15. Chai, X., Deng, L., Yang, Q., et al.: Test- cost sensitive Naive Bayes Classification. In: *IEEE International Conference on Data Mining (ICDM)* (2004)
16. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993)
17. Aggarwal, C.C., Yu, P.S.: A Survey of Uncertain Data Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 21(5), 609–623 (2009)
18. Qin, B., Xia, Y., Li, F.: DTU: A Decision Tree for Uncertain Data. In: Theeramunkong, T., Kijssirikul, B., Cercone, N., Ho, T.-B. (eds.) *PAKDD 2009. LNCS*, vol. 5476, pp. 4–15. Springer, Heidelberg (2009)
19. Tsang, S., Kao, B., et al.: Decision Trees for Uncertain Data. *IEEE Transactions on Knowledge and Data Engineering*, August 11 (2009)
20. Qin, B., Xia, Y., Prabhakar S., Tu, Y.: A Rule-based Classification Algorithm for Uncertain Data. In: *IEEE International Conference on Data Engineering* (2009)
21. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022–1027. Morgan Kaufmann (1993)

Direct Marketing with Fewer Mistakes

Eileen A. Ni and Charles X. Ling

Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
{ani,cling}@csd.uwo.ca

Abstract. Direct marketing is one of the most common and crucial business intelligence tasks. In direct marketing, the goal of an agent is to mine the right customers to market certain products, with the goal of making fewest mistakes. This data-mining problem, though similar to active learning in terms of allowing the agent to select customers actively, is, in fact, opposite to active learning. As far as we know, no previous data mining algorithms can solve this problem well. In this paper, we propose a simple yet effective algorithm called Most-Certain Learning (MCL) to handle this type of problems. The experiments show that our data-mining algorithms can solve various direct marketing problems effectively.

Keywords: direct marketing, minimal number of mistakes, most-certain learning.

1 Introduction

Direct marketing is one of the common and crucial business intelligence tasks. It is a process of identifying likely buyers to market products accordingly. Direct marketing has been increasingly used especially by the sectors of finance, insurance company and telecommunication [1,2].

In direct marketing, a learner (or an agent) must study customers' characteristics and needs, and actively select examples (customers) to market its products. This direct marketing problem is similar to traditional active learning in the sense that it allows to select examples (customers) actively to learn.

However, direct marketing problem cannot actually be solved by existing active learning algorithms in spite of the similarity. First of all, we argue that the goal of direct marketing is different from that of active learning. In direct marketing, obviously the goal of sales agents is to promote products to the "right" customers so that the promotion is more efficient [3], i.e., to minimize the number of *mislabeling* the examples (customers). However, existing active learning achieves just the opposite (see experiment section for the details).

Secondly, in direct marketing, the true label of a customer ("buying" or "not buying") is revealed after a learner (an agent) labels the customer as "buying" (and promotes). As a result, even if the label from the learner is incorrect, the current model still can be updated effectively with the true label. This is totally

different for active learning in that the update of a current model only depends on the labels obtained from oracles, even when they are noisy.

Lastly, since incorrect labeling does not influence the model updating, learners (sales agents) may label examples (customers) with their current knowledge (model) as usually investigating true labels (querying oracles) are very expensive or infeasible. If the labeling from the current model is correct (the agent promotes his products successfully), no cost occurs (as not querying oracle). This is different from that labels can only be obtained from labelers (oracles) in most existing active learning algorithms.

Due to the differences, existing active learning algorithms cannot solve this direct marketing problem well, and as far as we know, no previous data mining methods can achieve the goal.

In this paper, we propose a new, simple yet effective algorithm called *Most-Certain Learning (MCL)*, to solve the direct marketing and many similar problems. The basic idea of MCL is to select the next example that can be labeled by the current model with the highest certainty to learn. The rationale behind is that learning is a gradual process, and the uncertain examples can become certain ones with more certain examples being learned. As a result, the goal of minimizing the number of mistakes can be achieved.

Considering the requirements of real-world applications, we implement MCL further in two forms. One is that the learner must select and label all of the remaining examples as either positive or negative class, called *binary-class MCL*, or *MCL-b*. The goal of MCL-b thus is to minimize the number of mistakes of both classes. For example, a doctor usually must see and diagnose all patients as healthy or sick. The other type is that the learner only needs to identify one class of examples, called *single-class MCL*, or *MCL-1*. In direct marketing, an agent usually only cares about those customers who will likely buy the product (positive examples), and does not need to predict, approach, and verify customers who will unlikely buy the product (negative examples). In experiment section, we will see *MCL-1* applied to a market dataset.

We apply MCL on both UCI and KDD-Cup datasets. The experiment results show that, even though MCL is quite simple, still it works much better in minimizing the number of mistakes, compared to other learning strategies, including *most-uncertain sampling* (called *MUL* in this paper) which is a popular active learning strategy.

Furthermore, in the experiment we also discover another advantage of MCL: the learning process is much more stable than other learning strategies. This property is often important so that the learning behavior is more predictable. On the other hand, we will also show that MCL learns more slowly than MUL. That is, MCL needs to select more examples to learn a model as well as MUL does.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes Most-Certain Learning (MCL) paradigm and its implementation of two versions: MCL-b and MCL-1. We present the experiment results of

MCL-b and MCL-1 respectively in Section 4, and conclude our work and discuss the future work in Section 5.

2 Related Work

Direct marketing is a modern business activity with an aim to maximize the profit. As some complex, non linear models of customer behavior might be hidden in large amounts of data, data mining techniques are used to offer insight in the models [2]. However, most of the previous work focus on achieving some goals, such as improving the predicting accuracy, handling imbalanced data, and so on, with certain data mining techniques [1,2,3]. Our MCL algorithm focuses on how to control the marketing process and minimize the marketing cost. In addition, it can be used to solve many learning problems that are similar to direct marketing.

Direct marketing problem is an active process in the sense that it allows to select examples actively to learn. However, existing active learning [4,5] cannot solve it as their goals are different. Active learning is to minimize the number of labelings from oracles; while direct marketing is to minimize the number of mistakes in labeling examples.

Direct marketing problem is seemingly similar to agnostic active learning [6,7] as both of them are related to noise data. However, the noise in agnostic active learning comes from the oracles that provide labels, and the true label is hidden. The goal of agnostic active learning is to improve the sample efficiency. For our learning problem, the mislabelings come from the prediction of the current immature model and the true label will be revealed after the prediction, and the goal is also different.

Another work that is quite similar to MCL is self-directed learning [8,9,10]. Theoretically it studies the learning problem on simple classes of concepts, such as, disjunction, conjunction, k -DNF and so on, to minimize number of mistakes. However, the learning algorithm proposed must know and keep the set of all target concepts. It chooses the next example that has the greatest difference between the number of concepts that predict it differently (positive vs. negative labels). In a sense, the algorithm chooses the example that it predicts mostly certainly. However, the target concept class is often unknown, and nor is it feasible to keep all the concepts for a learning algorithm in real-world applications. As far as we know, there is no previous work in designing a practical learning algorithm that works well (i.e., making fewer mistakes) for direct marketing and similar problems.

3 Most-Certain Learning (MCL) Paradigm

Most-Certain Learning (MCL) is a high level learning paradigm, and can take any classifier that generates delicate probability prediction as its base learner, \mathcal{L} . MCL can be defined formally as follows. Let \mathcal{X} be the example set, and let \mathcal{C} be the concept class over \mathcal{X} . MCL works as follows: in each step, it chooses a new element (example) $x_i \in \mathcal{X}$ that can be labelled by the current learner \mathcal{L}

with the highest certainty. It then outputs label l_i given by \mathcal{L} and in response the true value $c_t(x_i)$ will be revealed, where $c_t \in \mathcal{C}$ denotes the target function. The learner will update its current model, m_i , with all the labelled examples when l_i does not agree with $c_t(x_i)$. The process continues until all the elements of \mathcal{X} are presented (MCL-b) or other stopping criteria are met (MCL-1). Let $M(MCL, c_t)$ denote the number of mistakes made by MCL, i.e., the total times of $l_i \neq c_t(x_i)$. The goal of MCL is to minimize the value of $M(MCL, c_t)$.

As mentioned, MCL requires a base learner \mathcal{L} that can generate prediction with delicate probability. In this paper, we take bagged decision trees [11] as the base learner for two reasons. First of all, it is easy to obtain an accurate probability of the prediction. The second one is that a large number of empirical studies in machine learning have shown that bagged decision trees classify more accurately compared to a single tree [12, 11].

One might wonder why MCL is able to reduce the number of mistakes efficiently. We explain it from version space perspective. Let \mathcal{X} be the example set, and let \mathcal{V} be the version space. For each $x_i \in \mathcal{X}$ ($i = 1, \dots, n$), its l_i given by the current learner is supported by \mathcal{V}_i ($\mathcal{V}_i \subseteq \mathcal{V}$). The most certain example selected by MCL is the one that is supported by the most concepts in \mathcal{V} , i.e., $\max(|\mathcal{V}_0|, \dots, |\mathcal{V}_i|, \dots, |\mathcal{V}_n|)$. Thus, if the label given is correct, there would be no mistakes happening; otherwise, the maximal set of concepts among $(\mathcal{V}_0, \dots, \mathcal{V}_i, \dots, \mathcal{V}_n)$ would be deleted from the version space \mathcal{V} , as the concepts are inconsistent with the example x_i . That is, the size of \mathcal{V} would reduce quickly, and fewer mistakes will happen for learning the concepts.

In the following subsection, we will present the implementation details of two types MCL, binary-class (MCL-b) and single-class (MCL-1) respectively.

3.1 MCL-b Learning Strategy

MCL-b is designed for the applications that need to label examples as either positive or negative, and its goal is to minimize the number of mistakes on labeling both positive and negative examples. MCL-b works under the framework of MCL as follows. It selects the most certain example to label with its current model, and updates its current model if the labeling is incorrect. This labeling and updating process iterates until all the examples are labelled.

To implement MCL-b, three technical issues deserve further discussion. The first issue is the selection of the most certain example. The second issue is the selection of the first example. The last issue is how to select an example when the labelled examples so far are of the same class.

The first issue, selection of the most certain example, is crucial. The most-certain example is the one that is predicted with the highest probability by the current model, i.e., the example that satisfies $\arg_{x_i \in \mathcal{X}} \max(\max(\text{prob}_{x_i}^+, (1 - \text{prob}_{x_i}^+)))$. $\text{prob}_{x_i}^+$ is the probability of predicting x_i as positive. In bagged decision trees, $\text{prob}_{x_i}^+$ is the number of decision trees that vote for positive out of the total number of decision trees.

The second issue, the selection of the first example, can be tricky, as the current model is empty. MCL-b scans the dataset and chooses the example that

appears with the highest frequency. If no example appears more than once, MCL-b chooses one example from the dataset randomly.

The last issue is how to choose the next example if labelled examples so far are of the same class, as a model built over the examples of the same class, say positive, will predict all the other examples as positive. In this paper, we use Euclidean distance [13] as heuristic information for selecting the next example. Assume that MCL-b has a positive example x_1 with nominal attribute values $\{1, 0, 0, 1\}$, we consider it as the center of positive, and the point that has the furthest Euclidean distance from x_1 (here the possible furthest Euclidean distance is 4) as the center of negative. The most certain example is the one that is closest to either of the two centers, and the example will be labelled with the label of the center. For example, if the nearest example from x_1 is y_u , and the Euclidean distance $d_{x_1 y_u} = 1$, and the furthest is y_v , and $d_{x_1 y_v} = 4$, MCL-b would consider y_v as the most certain example and label it as negative. The reason is that y_v has distance 0 to the negative center, which is closer than the distance of y_u to the positive center. This strategy can be formulated as follows.

$$y = \begin{cases} y_u, & \text{if } \sum_{x_i \in S} d_{x_i y_u} < |S| * m - \sum_{x_i \in S} d_{x_i y_v}; \\ y_v, & \text{otherwise.} \end{cases} \tag{1}$$

where, S is the set of labelled examples, and $|S|$ is the size of S , and m is the number of attributes. $d_{x_i y_u}$ is the Euclidean distance between x_i and y_u . y_u and y_v are the closest and furthest unlabeled examples to the current labelled examples respectively.

After introducing MCL-b, we will present the technical issues of MCL-1 in the following subsection.

3.2 MCL-1 Learning Strategy

MCL-1 is the other type of MCL. It is designed to label (retrieve) examples of one class, such as the customers who will buy the product in direct marketing or the movies one likes to watch in movie selection. Similar to MCL-b, MCL-1 selects the example that can be labelled by the current model with the highest certainty in each step, and updates its model with all labelled examples if the labeling is incorrect. However, as MCL-1 only cares about the examples of one class (to ease description, we assume the class that MCL-1 cares about is positive), the implementation techniques of MCL-1 are different from that of MCL-b on three aspects.

Firstly, MCL-1 chooses the most certain *positive* example to label. That is, MCL-1 chooses the example that satisfies $\arg \max(\text{prob}_{x_0}^+, \dots, \text{prob}_{x_i}^+, \dots, \text{prob}_{x_n}^+)$, where, $\text{prob}_{x_i}^+$ is the probability of labeling an example x_i ($x_i \in \mathcal{X}$) as positive by the current model. For instance, if two examples, x_1 and x_2 , are predicted as negative with probability 0.6 and positive with 0.5 respectively, MCL-1 will select x_2 to label, as it is more certain to be positive.

Secondly, MCL-1 chooses the example that has the shortest Euclidean distance to the *positive* center if all the labelled examples so far are of the same class.

Suppose that we have one labelled example x_1 and two unlabeled y_u , which is the closest to x_1 , and y_v , the furthest from x_1 . If x_1 is negative, MCL-1 will choose y_v and label it as positive; otherwise it will choose y_u and label it as positive.

Thirdly, what is the stop criterion for MCL-1? A good strategy would be to retrieve positive examples until the performance is not good enough. To measure the performance of retrieving positive examples, we use *F measure*, which is often used to evaluate the performance of Information Retrieval (IR) system [14,15]. More specifically, MCL-1 calculates F measure, product of precision and recall, in each step, and stops learning if the value of F measure reduces in three iterations successively (three-step lookahead strategy). After three successive steps of reduction, we expect that the value of F measure will not increase anymore with high certainty.

After presenting the most-certain learning (MCL) paradigm and its two versions, MCL-b and MCL-1, we will show the experiment results of MCL-b and MCL-1 respectively in the next section.

4 Experiment

In this experiment, to compare with MCL, we implement another three competitive learning strategies.

One strategy is most-uncertain sampling which is widely and successfully applied in active learning [4,16], and we call it *most uncertain learning (MUL)* in this paper. MUL always selects an unlabelled example that is labelled by the current model with the highest uncertainty. It is possible that MUL makes fewer mistakes in labeling the remaining examples, comparing to MCL, as many active learning works [4,16] have concluded that it builds a sophisticated model quickly. That is, MUL may make fewer mistakes during the later learning stage.

Another strategy is *MU2C*, which starts learning with the MUL strategy, and then switches to MCL in the middle of learning process. MU2C is implemented based on the idea that since MUL can build a good model quickly, and MCL is expected to minimize the number of mistakes. Thus, the combination of MUL and MCL may work better than either of the two strategies.

The last strategy is *Random*, which builds models by selecting examples randomly, and is used as a baseline for the comparison in the experiment.

The experiment is conducted on 10 UCI [17] datasets, including anneal, autos, breast_cancer, colic, diabetes, ecoli, glass, heart-h, sonar and vote, which are commonly used in the supervised learning research. 70% of each dataset is used as training data and 30% as test data. All the four learning strategies take bagged decision trees as their base learner, and are implemented based on the WEKA [18] source code. In the experiment, the t-test results are with 95% confidence.

4.1 MCL-b Experiment

In this experiment, we compare MCL with MUL, MU2C and Random on three aspects. Firstly, as minimizing the number of mistakes is the goal of the direct

marketing problem, the *number of mistakes* would be the most important one. Secondly, the performance of a good learning strategy should be stable. Thus, we use *volatility*, the standard deviation of the number of mistakes from running r times ($r=5$ in our experiment), to indicate how stable a learner is. More specifically, $volatility = (\sum_{i=0}^r (M(MCL, c_t)_i - \frac{1}{r} \sum_{i=0}^r M(MCL, c_t)_i)^2)^{1/2}$. The less the volatility, the more predictable the learning behavior is. The last aspect is *learning efficiency*, i.e., the number of examples that a learner needs to learn a model well. It indicates that if a learner is able to improve its learning model efficiently.

Comparison of number of mistakes. To observe the performance in terms of the number of mistakes, we run the learning strategies on each dataset for 5 times, and show the average number of mistakes on learning 1/4, 2/4, 3/4 and 4/4 of the training data respectively in Figure 1. We can see clearly that the mistakes of MCL-b on learning 1/4 data are much fewer than that of Random, MUL and MU2C, but with the increasing labelled data, the gap becomes smaller and smaller.

To make the comparison clearer, we summarize them further in Table 1. It shows that the average mistakes of MCL-b are 10%, 22% and 19% fewer than that of Random, MUL and MU2C respectively.

The t-test results also show that MCL-b wins the other three learning strategies on all the 10 datasets except that it ties with Random on three datasets (anneal, ecoli and vote) after the whole training data are learned. Thus, in general, the results in our experiment confirm the conclusion of theoretical work [19,9,20] that the number of mistakes can be reduced efficiently. However, it also indicates that mistakes cannot be reduced significantly on all datasets.

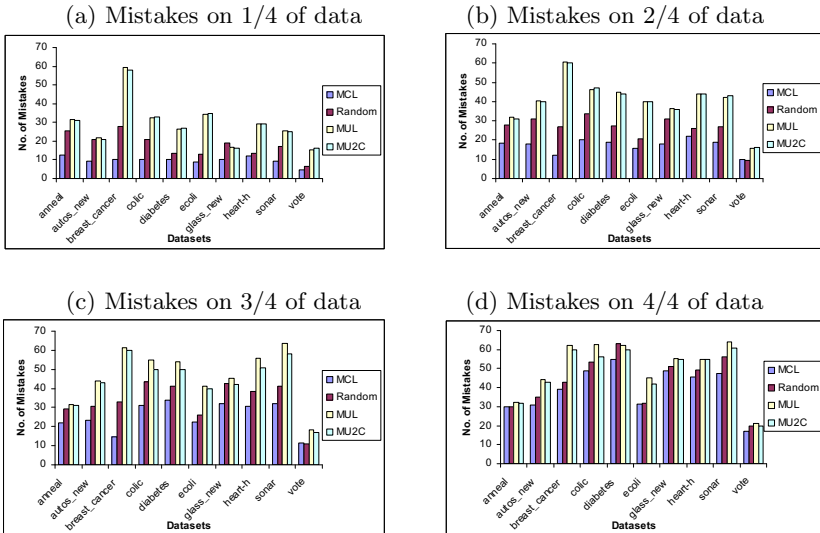


Fig. 1. The number of mistakes

Table 1. Gaps of mistakes between MCL and the others

	MCL/Random	MCL/MUL	MCL/MU2C
1/4 of data	55%	33%	34%
2/4 of data	66%	43%	43%
3/4 of data	75%	54%	57%
4/4 of data	90%	78%	81%

Why does the tie happen on the three datasets? The learning process on the first 12 examples in Figure 2 shows that the model built by Random can become sophisticated by learning a few examples. Sophisticate models indicates fewer mistakes in labeling the remaining examples, which leads to similar performance between Random and MCL-b.

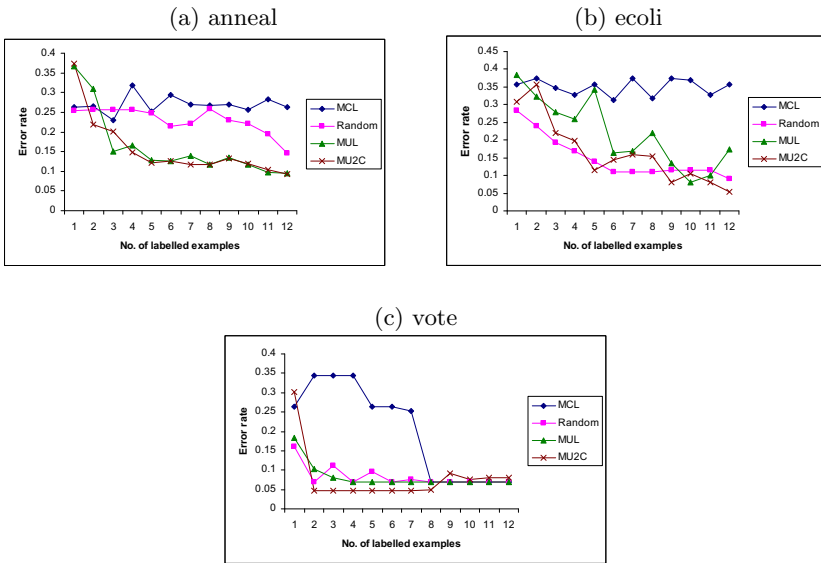


Fig. 2. Performances of models built on the first 12 examples

However, one might wonder why MUL makes more mistakes than Random and MCL-b, even though we can see that it improves its model even quicker than Random in Figure 2. To explain the reason well, we show the learning process on two datasets (due to the similar performances on the 10 datasets), anneal and autos, in Figure 3. The x and y axis are about the number of mistakes and the number of examples labelled. It shows that it still may make many mistakes after a good model is built, as shown in Figure 3. This is because that MUL always chooses the most uncertain example to predict. Even for a sophisticated model, mistakes still can happen on the most uncertain examples.

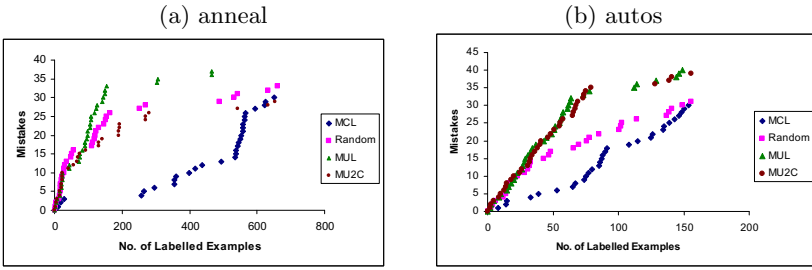


Fig. 3. Mistakes vs. labelled examples

Another interesting result is that MU2C makes almost the same mistakes as MUL. This is because MU2C makes most of the mistakes when using MUL strategy (i.e., before using MCL-b strategy), as MUL makes most of the mistakes at the beginning of learning process (See Figure 3).

Comparison of volatility. The volatilities of the four strategies are presented in Figure 4. It shows clearly that the volatility of MCL-b is significantly less than that of Random, MUL and MU2C on all the 10 datasets without exception. The reason is that MUL chooses the most-uncertain example in each step, which may affect the current model by a wide margin. On the other hand, MCL-b always selects the most-certain example, which may not be able to improve the current model much, but in a stable manner. The low volatility is important in real applications, as it indicates that the learning behavior is more predictable.

Comparison of learning efficiency. Active learning research has shown that MUL learns a good model quickly. On the other hand, MCL-b is expected to learn a model slowly. To compare the learning behavior of different strategies, we show the predicting error rate of the models on one datasets (due to the similar performance on the 10 datasets), autos, in Figure 5. The x and y axis are about the number of examples labelled and the prediction error rate of the current model on test data. Figure 5 shows clearly that the error rate of MUL drops dramatically, while the error rate of MCL-b reduces slowly, as expected. This indicates that the most uncertain examples are more informative and can improve the current model efficiently.

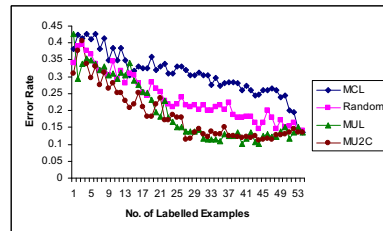
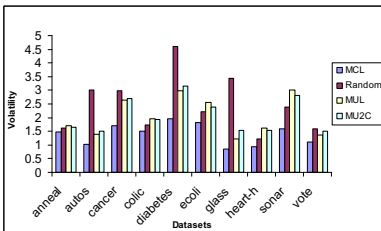


Fig. 4. Volatility of the Four Algorithms Fig. 5. Learning Process of dataset, autos

In summary, MCL-b does reduce the mistakes of labeling examples with its current model. Meanwhile, MCL-b produces more stable results from different runs than Random, MUL and MU2C. Thus, MCL-b is a better learning strategy for direct marketing and similar problems. However, MUL, which is considered to be a strong competitor to MCL-b, achieves the opposite in spite of its great learning efficiency (learning a good model with fewer examples).

4.2 MCL-1 Experiment

MCL-1 is to retrieve positive examples from a dataset. Accordingly we modify the learning strategies, Random, MUL and MU2C, so that they can be compared fairly. More specifically, Random selects positive examples randomly. That is, it labels an example if the prediction is positive; otherwise it reselects another one. MUL selects the most uncertain positive example. That is, the example that is labeled as positive but with the probability closest to 0.5. MU2C uses the MUL strategy at the first half of the learning process and then switches to MCL-1 strategy.

Due to the different goals, the measurement for MCL-1 is also different from MCL-b. Firstly, MCL-1 uses *F measure*, as mentioned in Section 3.2, to assess the performance of learning strategies, as the number of mistakes may not work well under the MCL-1 setting. For example, given a dataset with 20 positive examples, learning strategy *A* retrieves 10 examples as positive with 2 mistakes, and strategy *B* retrieves 20 examples with 4 mistakes. Strategy *A* would be better according to the number of mistakes, which, however, disagrees with the goal of MCL-1, i.e., retrieving more positive examples. Secondly, similar to MCL-b, *volatility* is also an important measurement in MCL-1. The difference is that the volatility in MCL-1 is the standard deviation of F measure.

As mentioned, MCL-1 stops learning if its F measure decreases on three steps successively. However, this lookahead strategy is not proper for Random, MUL and MU2C, as their F measure may still increase largely even after many steps of decrease. Thus, for them, we let the learning continues until all the examples are learned, and choose the peak value of F measure that they reach to compare with MCL-1.

In addition to the 10 UCI datasets, we conduct the experiment on a KDD-Cup marketing dataset, in which only the customers who probably will buy the product are concerned.

UCI Datasets. First of all, we show the F measure of MCL-1, Random, MUL and MU2C in Figure 6 on two datasets (due to similar results on all datasets), autos and sonar. In Figure 6, the curve of MCL-1 ends when F measure reduces successively for three steps (without learning the rest); while the curves of the other three strategies end at the peak values (learned all the rest but without showing them). We can see clearly that the F measure of MCL-1 still is much higher than that the other three strategies can get.

To be clearer, we summarize the comparison of F measure on the 10 UCI datasets in Figure 7. It shows the average value of F measure of 5 runs on each

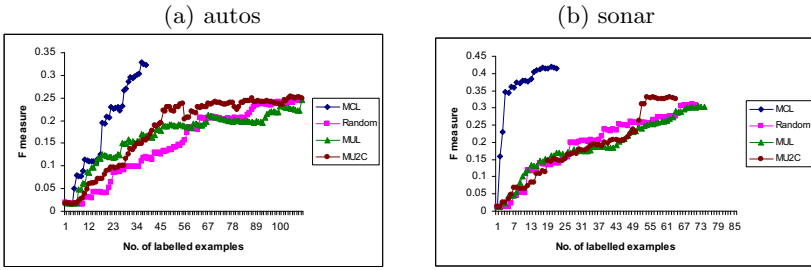


Fig. 6. F measure curves of MCL-1 (with three-step lookahead stop criterion) and Random, MUL and MU2C (maximal F measure)

dataset. The x axis is about the datasets and the y axis about F measure. On average, the F measure value of MCL-1 is about 1.4 to 1.7 times as much as that of the others. The t-test results of F measure also confirm that MCL-1 wins Random, MUL and MU2C on all the datasets without exception.

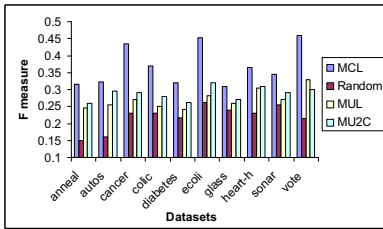


Fig. 7. Comparison of F measure on UCI datasets

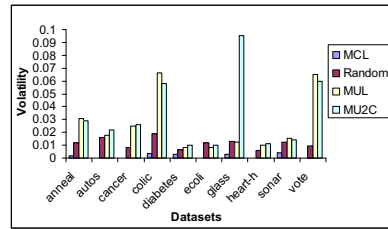


Fig. 8. Volatility of F measure on the 10 UCI datasets

Finally, we present the volatility of the four learning strategies in Figure 8. The x and y axis indicate datasets and volatility respectively. It shows that MCL-1 is extremely stable on all the 10 datasets, in which the volatilities of 5 datasets are almost zero. However, MUL is extremely volatile, even much more volatile than Random.

The performance of MCL-1 on UCI datasets is significantly better than the other three learning strategies on both F measure and volatility. Next we will show that its performance in real-world applications.

Real Application Data. Up-selling (marketing) data is a great option for observing the performance of MCL-1 on real-world applications. As in up-selling, an agent only cares about the customers who probably will purchase more expensive items. In this experiment, we run MCL-1 on up-selling (marketing) dataset (the small training set) from KDD-Cup 2009 [21], which has 50,000 examples. The first 190 attributes of the dataset are numerical and the rest 40 are categorical, and plenty of missing values are included. The class label is binary $\{+1, -1\}$, and “+1” indicates successful up-selling.

The F measure values of the four learning strategies (MCL-1, Random, MUL and MU2C) are shown in Figure 9. The results are consistent with the that on UCI datasets. In spite of the huge number of attributes and plenty missing values, MCL-1 still works significantly better than all the other learning strategies. Specifically, the F measure value of MCL-1 is about 35% higher than that of Random, and 25% higher than that of MUL and MU2C.

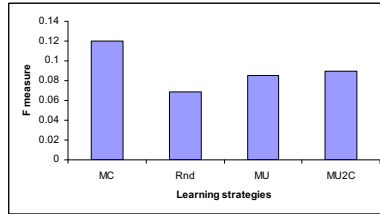


Fig. 9. F measure on the marketing data

However, the F measure value of MCL-1 is quite low (only about 0.12). To find out the reason, we check the precision and recall and find that the recalls for MCL-1 and MUL are quite good, close to 0.8 and 0.7 respectively (0.47 for Random), but the precisions are extremely low, 0.14 for MCL-1, and 0.09, 0.125 and 0.13 for Random, MUL and MU2C respectively. The low precision is caused by the extremely imbalance (more than 92% of the examples are negative) of the up-selling dataset.

The experiment results on both the UCI datasets and real marketing dataset show that MCL-1 is able to retrieve the examples of the class that we care about much more efficiently than the other learning strategies. Furthermore, MCL-1 produces much more stable results compared to the other strategies.

Our research is different from the competition of KDD Cup 2009. The competition is a traditional classification problem and concerns the prediction error rate and time efficiency. It is not for handling the problems proposed in this paper.

5 Conclusion and Discussion

In this paper, we proposed a practical learning paradigm, MCL, to fill the gap in mining the important class of tasks - direct marketing and many similar problems. MCL is implemented in two types, MCL-b and MCL-1, to handle binary- and single-class caring problems respectively. Our experiment results show that both MCL-b and MCL-1 outperform the other learning strategies significantly, including the strategy of active learning, MUL. Furthermore, MCL produces more stable results on all datasets. Another interesting phenomenon we noticed is that MUL makes more mistakes than MCL, even than Random, even though it is able to build a sophisticated learning model with the fewest labelled examples.

This paper makes important assumptions that we should visit in our future work so that we can understand the full benefits in real-world applications.

- In this paper, we assume that the mislabeling costs are evenly distributed. Varying costs and benefits of different examples may complicate the learning problem.
- In our work, we assume that a learner only labels an example with the prediction of its current model. Actually, a learner may prefer to query oracles if its confidence in the labeling with the current model is low or the cost for querying oracles is not high. That is, labeling an example by the current model or querying oracles should depend on the expected mistlabeling cost and querying cost.
- MU2C proposed in the experiment section is simple. Its learning strategy switches from MUL (aggressive learner) to MCL (conservative learner) in the middle of the learning process. Switching between aggressive and conservative dynamically may generate better learning performance.
- The experiment results on KDD-cup data show that the performance of MCL can be affected badly by the imbalanced data. Thus, further studies on imbalanced data is critical, as class distribution is often extremely imbalanced in marketing [1].

Despite of the limitations, we hope that this study provides an interesting topic for future studies to work on. Furthermore, we believe that our work can benefit part of the real-world applications immediately.

References

1. Ling, C., Li, C.: Data mining for direct marketing: Problems and solutions. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 73–79. Citeseer (1998)
2. Van Der Putten, P.: Data mining in direct marketing databases. In: Complexity and Management: A Collection of Essays. World Scientific, Singapore (1999)
3. Wong, K., Zhou, S., Yang, Q., Yeung, J.: Mining customer value: from association rules to direct marketing. *Data Mining and Knowledge Discovery* 11(1), 57–79 (2005)
4. Settles, B.: Active learning literature survey. *Machine Learning* 15(2), 201–221 (1994)
5. Cohn, D., Ghahramani, Z., Jordan, M.: Active learning with statistical models. Arxiv preprint cs/9603104 (1996)
6. Balcan, M., Beygelzimer, A., Langford, J.: Agnostic active learning. In: Proceedings of the 23rd International Conference on Machine Learning, 65–72. ACM (2006)
7. Dasgupta, S., Hsu, D., Monteleoni, C.: A general agnostic active learning algorithm. In: Advances in Neural Information Processing Systems, vol. 20, p. 2 (2007)
8. Ben-David, S., Kushilevitz, E., Mansour, Y.: Online Learning Versus Offline Learning. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 38–52. Springer, Heidelberg (1995)
9. Goldman, S.A., Rivest, R.L., Schapire, R.E.: Learning binary relations and total orders. *SIAM J. Comput.* 22(5), 1006–1034 (1993)

10. Rivest, R.L., Yin, Y.L.: Being taught can be faster than asking questions. In: COLT 1995: Proceedings of the Eighth Annual Conference on Computational Learning Theory, pp. 144–151. ACM, New York (1995)
11. Quinlan, J.: Bagging, boosting, and c4.5. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 725–730. AAAI Press (1996)
12. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 36(1-2), 105–139 (1999)
13. Shmueli, G., Patel, N., Bruce, P.: *Data mining for business intelligence*. John Wiley & Sons (2007)
14. Manning, C., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press (2008)
15. Singhal, A., Inc, G.: Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 24 (2001)
16. Tong, S.: *Active learning: theory and applications*. PhD thesis, Citeseer (2001)
17. Asuncion, A., Newman, D.: *UCI machine learning repository* (2007), <http://www.ics.uci.edu/~mllearn/mlrepository.html>
18. WEKA Machine Learning Project: Weka, <http://www.cs.waikato.ac.nz/~ml/weka>
19. Ben-David, S., Eiron, N., Kushilevitz, E.: On self-directed learning. In: COLT 1995: Proceedings of the Eighth Annual Conference on Computational Learning Theory, pp. 136–143. ACM, New York (1995)
20. Goldman, S.A., Sloan, R.H.: The power of self-directed learning. *Mach. Learn.* 14(3), 271–294 (1994)
21. Orange, F.T.C.: *Kdd cup 2009* (2009), <http://www.kddcup-orange.com/data.php>

Discovering Collective Viewpoints on Micro-blogging Events Based on Community and Temporal Aspects

Bin Zhao^{1,2}, Zhao Zhang¹, Yanhui Gu³, Xueqing Gong^{1,*},
Weining Qian¹, and Aoying Zhou¹

¹ Institute of Massive Computing,
East China Normal University, Shanghai, P.R. China
{zhzhang, xqgong, wqian, ayzhou}@sei.ecnu.edu.cn

² School of Computer Science and Technology,
Nanjing Normal University, Nanjing, P.R. China
zhaobin@nynu.edu.cn

³ Dept. of Information and Communication Engineering,
The University of Tokyo, Japan
guyanhui@tkl.iis.u-tokyo.ac.jp

Abstract. Towards hot events, microblogs usually collect diverse and abundant thoughts, comments and opinions in a short period. It is interesting and meaningful to find how users are thinking about such events. In this paper, we aim to mine collective viewpoints from micro-blogging messages for any given event. Since a user can post multiple messages in a discussion, a user may have multiple viewpoints on a given event. Also user viewpoints may change under the influence of external events, such as news releases and activities, as time goes by. These present challenging of extracting collective viewpoints. To address this, we propose a Term-Tweet-User (*TWU*) graph, which simultaneously incorporates text content, community structure and temporal information, to model user postings over time. We first identify representative terms from tweets, which constitute collective viewpoints. And then we apply Random Walk on *TWU* graph to measure the relevance between terms and group them into collective viewpoints. Finally, we evaluated our approach based on 817,422 tweets collected from Sina microblog, which is the biggest microblog in China. Experiments on the real dataset show the effectiveness of our model and algorithms.

Keywords: Random Walk, Graph clustering, Time decay function.

1 Introduction

A microblog is a popular Web 2.0 system, such as Twitter¹ and Buzz², which has received much attention in recent years. It allows users to post short messages,

* Corresponding author.

¹ <http://twitter.com>

² <http://www.google.com/buzz>

Table 1. Sample Tweets on “QQ vs. 360” Event

	Sample Tweet	Viewpoint
1	<p>解决同时使用360与qq的问题，找到目录“safebase”文件夹删除它，或改名后新建一个文本文档并改名为“safebase”，最后记得去除“txt”后缀名。</p> <p>To tackle the compatibility problem between 360 and QQ, locate the <u>folder</u> “safebase” and <u>delete</u> it, or create a new <u>text</u> document and <u>rename</u> it “safebase”, finally <u>remove</u> the <u>extension</u> “txt”.</p>	Technique
2	<p>我强烈鄙视 腾讯和奇虎伤害用户利益的行为</p> <p>I strongly <u>despise</u> <u>Tencent’s</u> and <u>Qihoo’s</u> acts harmful to <u>user interests</u>.</p>	Criticism

also known as *tweets*, which have up to 140 characters. However tweets cover a wide variety of content, ranging from breaking news, discussion, personal life, activities and interests. As a social network site, a microblog also helps users to interact with followers. Users can propagate their ideas rapidly, and retweet messages or hot links to their followers.

Micro-blogging is now developing into a broadcast medium expressing public opinion. Towards a hot event, microblogs usually collect diverse and abundant user thoughts, comments and opinions. *How do microblog users think about such event?* This is an interesting and practical problem.

As an example, consider discussions about hot events in microblogs. There is a popular business war (QQ vs. 360) between two firms on China Internet in 2010.³ China Internet firm Tencent shut down its popular instant messaging service QQ on computers installed with anti-virus software run by the company’s rival Qihoo 360, as a war between the two software giants escalated over the course of the two months. Two firms issued statements popping out on screens of millions of QQ and 360 users. The war had attracted widespread attention of microblog users. To support their viewpoints they posted all kinds of tweets, comments and opinions, including support, opposition and kidding and so on. A large amount of tweets with diverse viewpoints gathered in microblogs for such event in October 2010.

In the paper, our goal is *to identify collective viewpoints from massive tweets for any given event*. However, *what are collective viewpoints like? How to represent them?* Table 1 shows two tweets about “QQ vs. 360” event, which were posted by a specific user. The first tweet shows that the user tried to solve compatibility problem between two programs. It is from the viewpoint of technique. The second one shows criticism over both companies and dislike for the whole event. It is from the viewpoint of criticism.

So many individual viewpoints like above ones will converge into collective viewpoints. In order to mine them, we use representative words to depict every individual viewpoint and mine their clusters. For example, underlined words in Table 1 are representative words for presenting associated

³ http://www.chinadaily.com.cn/bizchina/2010-11/04/content_11501440.htm

viewpoints. {folder,delete,text,rename,remove,extension} is to depict the technical viewpoint, and {despise,Tencent,Qihoo,user interests} is to depict the critical viewpoint. Additionally, such example also shows the change from the technical to the critical viewpoint. Therefore, we assume that each user may have multiple viewpoints; furthermore, user viewpoints also may change as time goes by.

The closest research work is topic discovery, which has been studied extensively in the past [7,10,22,23]. They focus on document-level or sentence-level text mining. Usually they assume that only one topic is involved in one document or sentence. But this assumption is not suitable in real world, especially for viewpoint discovery. A tweet may support multiple viewpoints due to complicated language context. Therefore, existing methods can not be directly used for extracting viewpoints. In our work, viewpoint discovery focuses on capturing term-level correlations, which drive representative terms to coverage into collective viewpoints. Another similar research work is opinion mining [19,6,15]. They focus on sentiment words such as adjectives or adverbs. But mining viewpoints are not limited to such words.

Unlike a traditional medium, micro-blogging has some particular characteristics, such as short length, massive size, low quality, real-time nature, social network. Micro-blogging poses some challenges with regard to its characteristics. First, tweets are deficient in statistical and linguistic features due to short length. The existing methods for mining long texts are not suitable for microblogs. Second, user viewpoints are usually unclear, and sometimes will change as time goes by. *How to model user postings over time in order to capture them?* Finally, in the absence of the ground truth, evaluation of mining viewpoints is not trivial.

To summarize, our main contributions are as follows:

- We design a framework for mining viewpoints from massive tweets, and our proposed approach is language-independent in essence.
- We propose a Term-Tweet-User graph model which incorporates community and temporal information simultaneously for mining microblogs.
- We utilize a time decay function for help model user postings over time.
- In the absence of ground truth, we perform extensive experiments on real microblog dataset to demonstrate the effectiveness of our proposed approach.

The rest of the paper is organized as follows. Section 2 introduces the preliminary concepts and formulates the viewpoint discovery problem. Section 3 details random walk method based on *TWU* graph. Section 4 presents the experimental results on the real microblog dataset. Section 5 provides a review of related work. Finally, we conclude in Section 6.

2 Problem Statement

2.1 The Framework of Mining Viewpoints

In this section we introduce the proposed framework that aims to exploit viewpoints. The workflow consists of three consecutive phases, including *microblog*

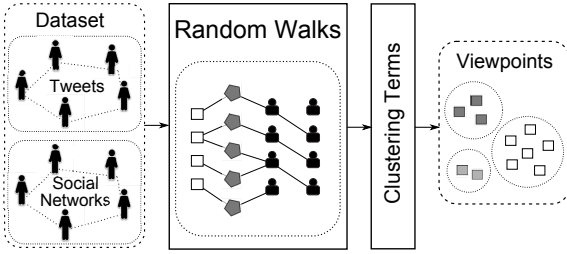


Fig. 1. Framework for Viewpoint Discovery

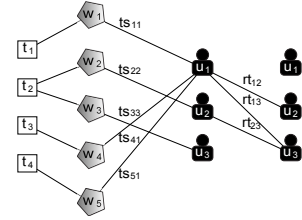


Fig. 2. Graph Representation

tweet collection, *term similarity measurement* and *viewpoint generation*, as shown in Fig. 1.

Microblog Tweet Collection Phrase. We design and develop a microblog crawling program, which posts keyword queries to microblog search engine and collects search results (i.e., tweet web pages), and then extract relationship between representative terms and tweets after word segmentation, and social network of associated users after analysing retweet interactions.

Term Similarity Measurement Phrase. We utilize text content, community structure and temporal information to construct a 4-partite graph, as shown in Fig. 2. Random walk method is posed on such graph in order to measure each term similarities with other ones.

Viewpoint Generation Phrase. With respect to each term, we obtain a ranking of similarities with other ones after the previous phrase. Clustering terms will be based on shared nearest neighbour (SNN). The result clusters are represented as all kinds of viewpoints, as mentioned in Section 1.

In next sections, we will focus on *Terms Similarity Measurement* phrase. Table 2 summarizes the major notations in this paper.

2.2 Term-Tweet-User Graph

In this paper, we construct a universal graph for mining microblogs in Fig. 2. The graph $G = (V, E)$ is a finite and node-labeled 4-partite graph. V may be divided into disjoint partite sets. Each edge in E connects two nodes from different partite sets; that is, there is no edge between two nodes in the same partite set. The graph G includes three types of objects (i.e., nodes): terms, tweets and users, which are denoted by node set T, W and U respectively. Node $t_i, w_i, u_i \in V$ denote three types of objects respectively. There are also three types of relations between different node sets, which are denoted by edge set E^{tw}, E^{wu} and E^{uu} respectively. Edge $e_{t_i w_j} \in E^{tw}$ represents that term t_i occurs in tweet w_j . Edge $e_{w_i u_j} \in E^{wu}$ represents that user u_j posts tweet w_i . And edge $e_{u_i u_j} \in E^{uu}$ represents that user u_i retweets u_j 's tweets. In a word, we obtain a *Term-Tweet-User* graph for mining microblogs.

Table 2. Main Notations Used in This Paper

Symbols	Description
G	4-partite graph (V, E)
T, W, U	Term node set, Tweet node set, User node set, $V = T \cup W \cup U$, $T = \{t_1, \dots, t_p\}$, $W = \{w_1, \dots, w_q\}$, $U = \{u_1, \dots, u_r\}$
$E^{tw} E^{wu} E^{uu}$	Term-Tweet, Tweet-User and User-User relationship set, $e_{v_i v_j} \in E$, $v_i, v_j \in V$, $E = E^{tw} \cup E^{wu} \cup E^{uu}$
$\mathbf{P} = [p_{ij}]$	the overall weighted matrix, $1 \leq i, j \leq (p + q + r)$
\mathbf{P}^N	the normalized matrix of \mathbf{P}
$\mathbf{P}^{tw} \mathbf{P}^{wt} \mathbf{P}^{wu} \mathbf{P}^{uw} \mathbf{P}^{uu}$	block matrixes of \mathbf{P} , $\mathbf{P}^{tw} = (\mathbf{P}^{wt})^T$
rt_{ij}	the retweet number from user u_i to user u_j , $u_i, u_j \in U$
ω	the weight function defined on edges
f	time decay function
α, c_1, c_2	time decay function parameters
β	the proportion of time weight of a user, $\beta \in [0, 1]$
γ	$1 - \gamma$ is the restart probability for random walks
$\mathbf{S} = [s_{ij}]$	$p \times (p + q + r)$ term ranking matrix
$\mathbf{s}_i = [s_{ij}]$	$1 \times (p + q + r)$ ranking vector, s_{ij} is the relevance score of v_j w.r.t. v_i
\mathbf{e}_i	$1 \times (p + q + r)$ starting vector, the i^{th} element 1 and 0 for others
$\mathbf{T} = [ts_{ij}]$	$r \times q$ time-stamp matrix
ts_{ij}	time-stamp of tweet w_j posted by user u_i , $u_i \in U$, $w_j \in W$
\mathbf{Z}	$r \times q$ matrix of time weights
VP	viewpoint set, $VP = \{VP_1, VP_i, \dots, VP_n\}$
VP_i	the i^{th} viewpoint, $VP_i \in VP$
t_{ij}	the j^{th} term in viewpoint $VP_i \subseteq T$

2.3 Problem Definition

Definition 1. (VIEWPOINT) *Towards a specific event, multiple viewpoints $VP = \bigcup_{i=1}^m VP_i$ may emerge. Each viewpoint VP_i can be depicted by a few terms, $VP_i = \bigcup_{j=1}^m t_{ij}$, $t_{ij} \in T$.*

With the above notations, our problem can be formally defined as follows:

Problem 1. (VIEWPOINT DISCOVERY)

Given: TWU 4-partite graph G .

Goal: To exploit collective viewpoints VP .

A certain user may post several tweets in a period of time. These tweets may support multiple viewpoints. For example, in Table 1 the user criticized Qihoo and Tencent very strongly while tackling the compatibility problem between 360 and QQ. On the other hand, his viewpoints changed as time goes by. Hence, the key problem is *how to model user postings over time for mining viewpoints?* The answer is TWU graph model, which incorporates main aspects of user postings (i.e., community and temporal aspects). Based on such model, we will make the terms supporting similar viewpoints gather together in one cluster. In the next section, we will detail our proposed methods for the problem.

3 Mining Viewpoints on *TWU* Graph

In this section, we will mine collective viewpoints based on *TWU* graph from massive tweets. We first present the overview of our approach. Then we describe how to model community and temporal features based on *TWU* graph. We apply Random Walk on such graph to measure term relevance and group representative terms into viewpoints.

3.1 Our Solution

1. To extract representative terms from tweets.
2. To construct *TWU* graph by leveraging text, community structure and temporal information.
3. To apply random walk on *TWU* graph to compute the ranking of its nearest neighbours for each representative term.
4. To divide terms into clusters using shared nearest neighbour.

Towards *step* (1), we can use existing methods such as TF-IDF to measure term importance and extract representative terms. Towards *step* (4), we utilize existing clustering algorithms to implement it. In this paper, we mainly focus on *step* (2), (3).

3.2 Modeling Community and Temporal Aspects

Due to short length, tweets can provide limited textual information. Traditional text representation methods, such as "bag of words", may result in poor quality of mining. However tweets are able to provide other features like community structure and temporal information. Hence, leveraging them will achieve better results.

Owing to massive amount of community, it is infeasible to collect the entire community graph. Usually users remark interesting tweets by means of retweets when they are interested in hot tweets and the following's tweets. The frequency of retweet interactions between users is proportional to their relevance. For ease of illustration, we present an example in Fig. 3. User u_3 should be more closer than user u_2 w.r.t. user u_1 . This shows that retweet interactions include potential community information. Therefore, we utilize retweets to construct an approximate community graph for modeling community aspect.

Compared to community aspect, modeling temporal aspect is not straightforward. For example, user u_1 posted three tweets in a short period of time in Fig. 3. The gap between tweet w_1 and tweet w_2 is one day, and the gap between tweet w_1 and tweet w_3 is seven days. We observe that the viewpoint of a certain user remains relatively steady in the short term. In other words, the closer two tweets are, the greater the relevance between them is. Therefore, w_2 is closer than w_3 at the semantic level w.r.t. w_1 . In next sections, we will thoroughly discuss how to compute temporal weight based on *TWU* graph.

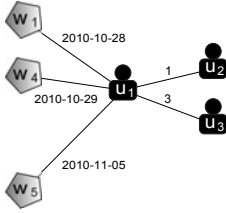


Fig. 3. An Example of *TWU* Model

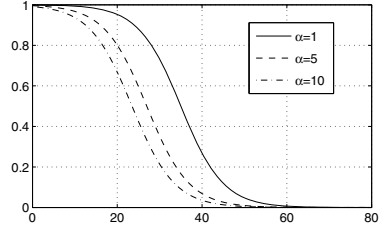


Fig. 4. Logistic Functions with Different Parameters, $c_1 = 5, c_2 = 7$

3.3 Random Walk on *TWU* Graph

3.3.1 Random Walk Method

Random walks with restart is defined as Equation (11), which is applied on *TWU* graph. The equation can provide proximity measurement. s_i represents the proximity vector w.r.t. node t_i . s_{ij} represents the relevance score of node t_j w.r.t. node t_i , $1 \leq j \leq p$.

$$s_i = \gamma s_i P^N + (1 - \gamma) e_i \tag{1}$$

There are two widely used ways (*PreCompute* and *OnTheFly*) to solve random walks with restart [16]. *PreCompute* method requires pre-computation and is suitable for queries of node proximity in the on-line response time. *OnTheFly* method does not require pre-computation and additional storage cost. Its response time depends on the iteration number and the number of edges. Therefore, *OnTheFly* method is more suitable especially if the graph is massive. In this paper, we prefer *OnTheFly* method for computing the relevance scores of nodes.

3.3.2 Measuring Temporal Relevance

Time decay functions are normally used for defining weight functions based on time-stamps. Viewpoints are usually sensitive to time and vary with time. Hence, a time decay function can reflect the relevance of different tweets. As we mentioned before, a user can post several tweets about a certain event. The relevance of tweets separated by a long time is very different from that of a short time. Therefore, the probability of tweets sharing the same viewpoint is higher in the short term.

In our proposed method, we choose a logistic function to model user postings over time. The function is defined as Equation (2), which is a monotonic decreasing function. The value of the function is in the range (0,1) and reduces with the time interval. Fig. 4 shows three logistic functions with different parameter α values. We can observe S-shaped curve of the logistic functions. The gradient of the curve at the data point close to zero is gentler, and the gradient of the curve at the middle data point is steepest.

$$f(x) = \frac{1}{1 + \alpha * e^{(x/c_1 - c_2)}} \tag{2}$$

where α can adjust the curve shape of the logistic function. The smaller α is, the smaller variation range of the function is at the initial stage. c_1 and c_2 determine the time duration. x represents duration between time-stamps.

In order to compute the time weights for edges with time-stamps, we firstly define a time-stamp matrix as:

$$\mathbf{T} = [ts_{ij}]_{r \times q} \quad (3)$$

where ts_{ij} is edge $e_{u_i w_j}$'s time-stamp, otherwise $ts_{ij} = 0$.

Then, $\forall w_k \in W$, its duration matrix is defined as:

$$\Delta \mathbf{T}_k = [\Delta ts_{ij}^k]_{r \times q}, \quad 1 \leq k \leq q \quad (4)$$

where Δts_{ij}^k is as follows:

$$\Delta ts_{ij}^k = \begin{cases} |ts_{ij} - ts_{ik}| & 1 \leq i \leq r, \quad e_{u_i w_j} \in E^{uw} \\ \infty & otherwise \end{cases} \quad (5)$$

where Δts_{ij}^k represents duration between tweet w_j and tweet w_k w.r.t. user u_i . The time function is as follows:

$$f(\Delta ts) = \begin{cases} 1 & \Delta ts = 0 \\ 0 & \Delta ts > 2 * c_1 * c_2 \\ \frac{1}{1 + \alpha * e^{(\Delta ts / c_1 - c_2)}} & otherwise \end{cases} \quad (6)$$

$$\omega(e_{u_i w_j}) = \sum_{k=1}^q \omega_k(e_{u_i w_k}) = \sum_{k=1}^q f(|ts_{ij} - ts_{ik}|), \quad \forall u_i \in U, \quad \forall w_j \in W \quad (7)$$

According to the theory of random walk, random particle may transmit to edges with high weights [16]. In our example in Fig. 3. The particle that starts from node w_1 transmits to other tweet nodes via node u_1 . The probability of the transmission is proportional to their edge weights. In our proposed methods, the contribution of edge $e_{u_1 w_1}$ is determined by the duration between ts_{11} and other time-stamps. The edges whose time-stamps are closer to ts_{11} should get more from edge $e_{u_1 w_1}$. Thus, edge $e_{u_1 w_4}$ receives a larger contribution from $e_{u_1 w_1}$ than edge $e_{u_1 w_5}$. In addition to $e_{u_1 w_1}$, other edges can also contribute. Therefore, we can obtain the overall weight of a certain edge after aggregating others' contributions. Finally, we perform the same operation on every edge and obtain a weight matrix Z .

$$\mathbf{Z}_k = [f(\Delta ts_{ij}^k)]_{r \times q}, \quad 1 \leq k \leq q \quad (8)$$

$$\mathbf{Z} = \sum_{k=1}^q \mathbf{Z}_k = [z_{ij}]_{r \times q} = [\sum_{k=1}^q f(\Delta ts_{ij}^k)]_{r \times q} \quad (9)$$

3.4 Our Algorithm

Algorithm 1 gives the overview of our proposed method. We first construct the graph representation of *TWU* model from the original raw dataset (step 1-4). Then (step 5-16), we build the weighted matrix \mathbf{P} , which consists of five block matrixes: *Term-Tweet* block matrix \mathbf{P}^{tw} , *Tweet-Term* block matrix \mathbf{P}^{wt} , *Tweet-User* block matrix \mathbf{P}^{wu} , *User-Tweet* block matrix \mathbf{P}^{uw} and *User-User* block matrix \mathbf{P}^{uu} . The first three block matrixes can be computed from the adjacency matrix \mathbf{W} (step 6-8). Block matrix \mathbf{P}^{uw} can be computed using logistic functions (step 9-12). Block matrix \mathbf{P}^{uu} can be computed from the retweet frequencies (step 13-15). Then (step 17), we will normalize matrix \mathbf{P} . Finally (step 18-24), Equation (1) of random walk will be iterated until convergence. In the algorithm, convergence should occur if the weight distribution of edges does not change in each iteration.

$$\mathbf{P} = [p_{ij}]_{(p+q+r) \times (p+q+r)} = \begin{bmatrix} \mathbf{0} & \mathbf{P}^{tw} & \mathbf{0} \\ \mathbf{P}^{wt} & \mathbf{0} & \mathbf{P}^{wu} \\ \mathbf{0} & \mathbf{P}^{uw} & \mathbf{P}^{uu} \end{bmatrix} \tag{10}$$

In our example in Fig. 3, *User* nodes connect two kinds of nodes: *User* and *Tweet*. We should balance two kinds of weights on time-stamp and user relationship. A simple solution is to set the parameter β .

$$p_{ij} = \omega(v_i, v_j) = \begin{cases} 1 & e_{v_i v_j} \in E^{wt}, e_{v_i v_j} \in E^{tw}, e_{v_i v_j} \in E^{wu} \\ \beta * z_{ij} & e_{v_i v_j} \in E^{uw} \\ (1 - \beta) * rt_{ij} & e_{v_i v_j} \in E^{uu} \\ 0 & otherwise \end{cases} \tag{11}$$

while β is a weighting factor.

$$\mathbf{P}^N = [p_{ij}^N]_{(p+q+r) \times (p+q+r)} = \left[\frac{p_{ij}}{\sum_{j=1}^{p+q+r} p_{ij}} \right]_{(p+q+r) \times (p+q+r)} \tag{12}$$

In Algorithm 1, there are several ways to normalize the weighted matrix. Our proposed algorithm uses row normalization that is a most natural method to normalize matrix \mathbf{P} . Correspondingly, we take the normalized matrix \mathbf{P}^N as the input.

4 Experiments

4.1 Dataset Description

We obtained the real dataset from Sina microblog⁴, which has been developing rapidly and greatly in China. According to Sina⁵, it has more than 50 million users, and is adding 10 million new users per month in 2010. We collected 817,422

⁴ <http://www.weibo.com>

⁵ <http://www.sina.com>

Algorithm 1. Compute the Proximity Vectors of Terms

Input: TWU 's Dataset, β, γ
Output: Term Ranking Vector $\mathbf{s}_i, 1 \leq i \leq p$

- 1 $n \leftarrow p + q + r$;
- 2 Construct adjacent matrix \mathbf{W} using TWU 's Dataset, $\mathbf{W} \leftarrow [w_{ij}]_{n \times n}$;
- 3 Construct time-stamp matrix \mathbf{T} using TWU 's Dataset, $\mathbf{T} \leftarrow [ts_{ij}]_{r \times q}$;
- 4 Construct retweet matrix \mathbf{R} using TWU 's Dataset, $\mathbf{R} \leftarrow [rt_{ij}]_{r \times r}$;
- 5 Initialize the weighted matrix: $\mathbf{P} \leftarrow \mathbf{0}_{n \times n}$;
- 6 **for** $i \leftarrow 1$ **to** $p + q$ **do**
- 7 **for** $j \leftarrow 1$ **to** n **do**
- 8 $p_{ij} \leftarrow w_{ij}$;
- 9 **for** $i \leftarrow 1$ **to** r **do**
- 10 **for** $j \leftarrow 1$ **to** q **do**
- 11 **if** $ts_{ij} \neq 0$ **then**
- 12 $p_{ij} \leftarrow \beta * \sum_{k=1}^q f(|ts_{ij} - ts_{ik}|)$;
- 13 **for** $i \leftarrow 1$ **to** r **do**
- 14 **for** $j \leftarrow 1$ **to** r **do**
- 15 $p_{(i+p+q)(j+p+q)} \leftarrow (1 - \beta) * rt_{ij}$;
- 16 $\mathbf{P} \leftarrow [p_{ij}]_{n \times n}$;
- 17 Normalize \mathbf{P} to obtain \mathbf{P}^N ;
- 18 **foreach** term $t_i \in T$ **do**
- 19 $\mathbf{s}_i \leftarrow \mathbf{e}_i$;
- 20 **repeat**
- 21 $\mathbf{s}_i \leftarrow \gamma \mathbf{s}_i \mathbf{P}^N + (1 - \gamma) \mathbf{e}_i$;
- 22 **until** convergence ;
- 23 $\mathbf{S} = [\mathbf{s}_i]_{p \times 1}$;
- 24 **return** \mathbf{S} ;

tweets about ‘‘QQ vs. 360’’ war, which were posted by 382,622 users. The dataset was collected using Sina microblog search engine. The duration is from October 27th to November 10th in 2010. We will demonstrate the effectiveness of our proposed methods through ‘‘QQ vs. 360’’ case.

4.2 Evaluation Metric

Since no ground truth is available for microblog datasets. We evaluate the quality of clustering results using cohesion. Given a set of clusters $C = \{C_1, C_2, C_3, \dots, C_n\}$, we measure the cohesion as follows:

$$Cohesion(C) = \sum_{i=1}^n \omega_i Cohesion(C_i) \quad (13)$$

Where ω_i is the weight of C_i . $Cohesion(C_i) = \sum_{x \in C_i, y \in C_i} Proximity(x, y)$. The overall cluster cohesion $Cohesion(C)$ is the weighted sum of all cluster cohesions.

We measure the cohesion of the individual cluster in terms of the proximity of pairwise terms in the cluster. In the absence of reliable semantic knowledge

Table 3. Statistics of Postings

Time Interval	# Hours	Percent of Users
1	46.24	61.67%
2	31.73	29.40%
3	26.15	16.68%

bases like WordNet, we quantify the proximity of terms using *pointwise mutual information* (*PMI*). The *PMI* formula is defined as: $PMI(x, y) = \log(\frac{p(xy)}{p(x)p(y)})$. The more relevant two terms are, the larger the *PMI* is. Therefore, the higher cohesion values are better.

4.3 Comparison Methods

In recent years, there are some random walk methods considering temporal aspect in graph mining, such as temporal recommendation algorithm [21] (*IPF*) and time-stamp clustering algorithm [18] (*T3*). These algorithms treat time-stamps as nodes in the graph. But our proposed algorithm (*RWT*) utilizes time weight functions to increase the effectiveness of random walks.

In order to demonstrate the effectiveness of our proposed methods, we design two algorithms, random walk treating time-stamps as nodes (*RWTO*) and random walk without temporal data (*RWNT*), to compare with *RWT* in the experiments. The results of *RWTO* and *RWNT* are used as baselines to examine whether temporal data can improve the effectiveness and how to utilize temporal data effectively.

As mentioned in Section 3.3.2, we obtains the multiple tweets of the same user to measure the similarity between two tweets using the time decay functions. The parameter setting is dependent on the statistics of user behaviours. Table 3 shows the statistics of user postings. The average gap between the 1st posting and the 2nd posting is 46.24 hours. The average gap between the 2nd posting and the 3rd posting is 31.73 hours. The average gap between the 3rd posting and the 4th posting is 26.15 hours. The whole duration is about one week. And 61.67% users joined in the discussion about “QQ vs. 360” war more than twice. As a result, we empirically set the value $\alpha = 10$, $\beta = 0.5$, $c1 = 1.5$ and $c2 = 6$.

We extract 2,000 terms from the dataset, which will be included in *TWU* graph. In *Viewpoint Generation* phrase in section 2.1, we adopt *DBSCAN* clustering algorithm with shared nearest neighbour (*SNN*) for term clustering. We evaluate the cohesions of *RWT*, *RWNT* and *RWTO* by setting different *DBSCAN*’s parameters: *Eps* and *MinPts*. In our experiments, *Eps* is *SNN* similarity, and is set from 7 to 10, and *MinPts* is from 5 to 7. Fig. 5 shows that our proposed algorithm *RWT* outperforms *RWNT* and *RWTO* in all cases.

Consider an extreme case, if every cluster contains two terms being most relevant, the overall cohesion must be highest. But the algorithm may generate too many viewpoint clusters. In such case, the experimental results are not meaningful. Therefore, besides the cohesion, we should examine the quantity of clusters.

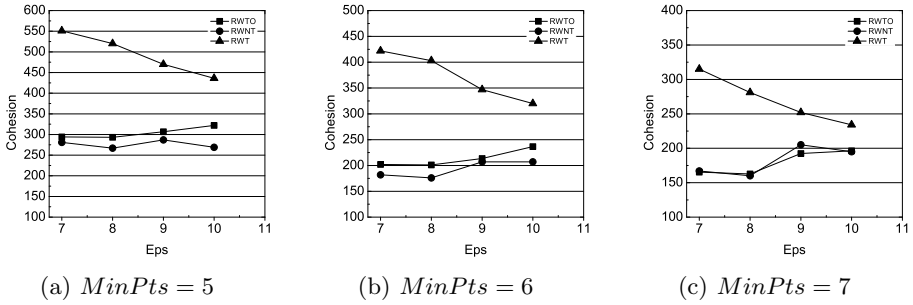


Fig. 5. Cohesion of *RWTO*, *RWNT* and *RWT* with different *Eps*

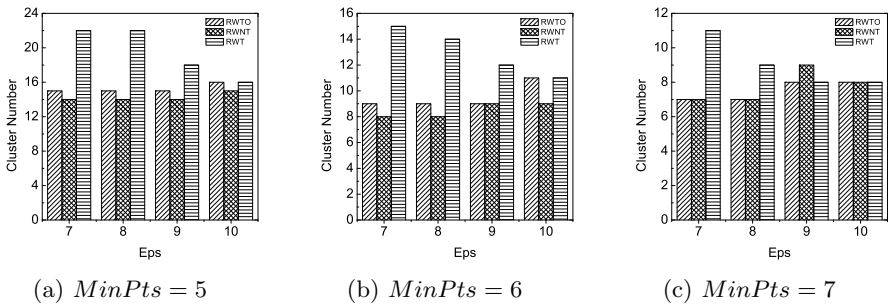


Fig. 6. Cluster Number of *RWTO*, *RWNT* and *RWT* with different *Eps*

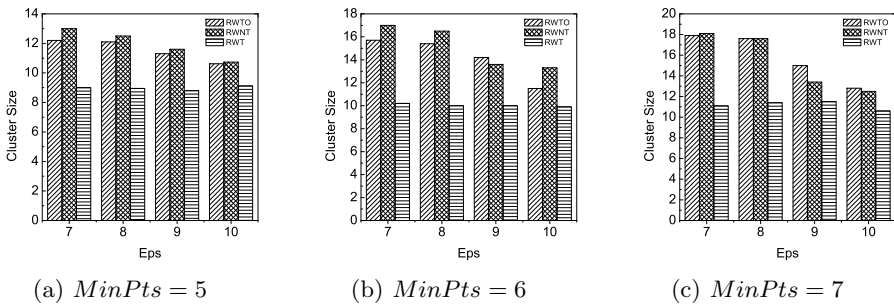


Fig. 7. Cluster Average Size of *RWTO*, *RWNT* and *RWT* with different *Eps*

Fig. 6 and 7 show the statistics of clusters generated by three algorithms with different parameters. The number and size of clusters generated by all algorithms are appropriate and rational. This proves that cohesion is an effective evaluation of three algorithms in our experiments.

Finally, compared to the other algorithms, *RWT* considers more features like temporal aspect and community structure. Thus, *RWT* achieves the satisfactory performance in the clustering of short texts.

5 Related Work

In this section, we introduce related work on topic summarization at first, and then describe random walk methods and time decay function.

Topic Summarization. The most closest work is topic summarization. Hierarchical summarization using agglomerative clustering [22,23] is one of the most used methods. Document clustering is used to achieve topic summarization. Yang et al. [22] apply hierarchical and non-hierarchical document clustering algorithms to a corpus of news stories, focusing on the exploitation of both content and temporal information. Zhao et al. [23] propose constrained agglomerative algorithms that combine the features of both partitional and agglomerative algorithms for text clustering. On the other hand, some methods utilize sentences to achieve topic summarization. Hu et al. [7] aim to extract representative sentences from a blog post that best represent the topics discussed among its comments. Li et al. [10] propose a novel method that incorporates novelty, coverage and balance requirements into a sentence ranking probability model for producing summaries highly relevant to the topic. Note that multiple viewpoints actually may be involved in one tweet because of complicated human sentiments. Thus, these methods are not suitable for viewpoint summarization.

Since short texts have few semantic and statistical features, [8,3] improve the accuracy of short text clustering by enriching their representation with additional features from Wikipedia or WordNet. However, these methods are not suitable for Chinese tweet clustering for lack of reliable knowledge bases like WordNet.

Besides content information, [4,13] take structural information into account to achieve topic discovery. Carenini et al. [4] propose a new framework for email summarization, which utilizes clue words to combine the content and the structure of the quotation graph. Qamra et al. [13] propose a Content-Community-Time graph that can leverage the content of entries, their timestamps, and the community structure of the blogs, to automatically discover stories. However, the methods can't be directly applied to mining microblogs due to its characteristics, such as the style of user interaction and posting.

Also, there are a lot of applications about twitter. O'Connor et al. [11] introduce 3 regression machine learning models (Direct model, Two-step pipeline model and Two-step blended model) to identify controversial events using Twitter. Popescu et al. [12] present TweetMotif, an exploratory search application for Twitter. TweetMotif provides the user a concise summary of themes and variation in the query subcorpus, then allow the user to navigate to individual topics to see their associated messages, and allow recursive drilldown. Sakaki et al. [14] investigate the real-time interaction of events such as earthquakes in Twitter, devise a classifier of tweets and subsequently produce a probabilistic spatio-temporal model to monitor tweets and to detect a target event.

Random Walk Methods. Random walk is one of the most widely used proximity measurement on graphs. Some research work mainly focus on random walk algorithms itself. Tong et al. [16] aim to improve the efficiency of random walk algorithms for large graphs by exploiting two important properties, linear

correlations and block wise community-like structure. Zhou et al. [24] propose a novel graph clustering algorithm, SA-Cluster, based on both structural and attribute similarities through a unified distance measure.

There are some research work considering temporal aspect in graph mining. Tong et al. [17] propose pTrack and cTrack methods to monitor the centrality of an individual node and the proximity of two nodes or sets of nodes. Tong et al. [18] propose algorithm T3 and MT3 to find patterns in a collection of time-stamped, complex events. However, such graph models are not suitable to our problem. Time-stamps are shared by all items as global patterns. The correlation based on such models is not meaningful to our problem. Xiang et al. [21] propose Session-based Temporal Graph (STG) which simultaneously models users' long-term and short-term preferences over time. Based on the STG model framework, they propose a recommendation algorithm Injected Preference Fusion (IPF) and extend the personalized Random Walk for temporal recommendation. Note that user preferences are different from the behaviours of user postings.

Also there are a lot of applications about random walk. Backstrom et al. [1] develop an algorithm based on Supervised Random Walks that naturally combines the information from the network structure with node and edge level attributes. Baluja et al. [2] present a novel method based upon the analysis of the entire user-video graph to provide personalized video suggestions for users. Keikha et al. [9] extract most relevant blogs for each query by applying a random walk to a blogosphere graph. Wijaya et al. [20] explore Pagerank's applications to sentiment analysis and opinion mining. They propose various techniques using collocation and pivot words to extract a weighted graph of terms from user reviews and to account for positive and negative opinions.

Time Decay Functions. There has been some work using time decay functions. Ding et al. [5] propose an algorithm to compute the time weights for different items. The algorithm take changes in user interest into consideration. Yang et al. [22] utilize time decay function for on-line detection of novel events.

6 Conclusions

In this paper, we propose a novel approach to mine collective viewpoints in microblogs. On account of the characteristics of microblogs, we propose a graph-based model named Term-Tweet-User (*TWU*), which incorporates community structure and temporal information, to model user postings over time. We applied Random Walk on *TWU* to measure the relevance between representative terms, and group them into collective viewpoints. The extensive experiments on the real dataset confirm the effectiveness of our proposed method over other existing ones.

Acknowledgements. This work is partially supported by National Science Foundation of China under grant numbers 60833003, 61070051, and 60803022, National Basic Research (973 program) under grant number 2010CB731402, and National Major Projects on Science and Technology under grant number 2010ZX01042-002-001-01.

References

1. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: WSDM (2011)
2. Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., Aly, M.: Video suggestion and discovery for youtube: taking random walks through the view graph. In: WWW (2008)
3. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: SIGIR (2007)
4. Carenini, G., Ng, R.T., Zhou, X.: Summarizing email conversations with clue words. In: WWW (2007)
5. Ding, Y., Li, X.: Time weight collaborative filtering. In: CIKM (2005)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD (2004)
7. Hu, M., Sun, A., Lim, E.-P.: Comments-oriented blog summarization by sentence extraction. In: CIKM (2007)
8. Hu, X., Sun, N., Zhang, C., Chua, T.-S.: Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: CIKM (2009)
9. Keikha, M., Carman, M.J., Crestani, F.: Blog distillation using random walks. In: SIGIR (2009)
10. Li, X., Shen, Y.-D., Du, L., Xiong, C.-Y.: Exploiting novelty, coverage and balance for topic-focused multi-document summarization. In: CIKM (2010)
11. O'Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. In: ICWSM (2010)
12. Popescu, A.-M., Pennacchiotti, M.: Detecting controversial events from twitter. In: CIKM (2010)
13. Qamra, A., Tseng, B.L., Chang, E.Y.: Mining blog stories using community-based and temporal clustering. In: CIKM (2006)
14. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW (2010)
15. Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., Su, Z.: Hidden sentiment association in chinese web opinion mining. In: WWW (2008)
16. Tong, H., Faloutsos, C., Pan, J.-Y.: Fast random walk with restart and its applications. In: ICDM (2006)
17. Tong, H., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Proximity tracking on time-evolving bipartite graphs. In: SDM (2008)
18. Tong, H., Sakurai, Y., Eliassi-Rad, T., Faloutsos, C.: Fast mining of complex time-stamped events. In: CIKM (2008)
19. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: ACL (2002)
20. Wijaya, D.T., Bressan, S.: A random walk on the red carpet: rating movies with user reviews and pagerank. In: CIKM (2008)
21. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long-and short-term preference fusion. In: KDD (2010)
22. Yang, Y., Pierce, T., Carbonell, J.G.: A study of retrospective and on-line event detection. In: SIGIR (1998)
23. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: CIKM (2002)
24. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. PVLDB 2(1) (2009)

Discriminatory Confidence Analysis in Pattern Mining

Russel Pears¹, Yun Sing Koh², and Gillian Dobbie²

¹ School of Computing and Mathematical Sciences, AUT University, New Zealand

rpears@aut.ac.nz

² Department of Computer Science, University of Auckland, New Zealand

{ykoh,gill}@cs.auckland.ac.nz

Abstract. The field of association rule mining has long been dominated by algorithms that search for patterns based on their frequency of occurrence in a given dataset. The birth of weighted association rule mining caused a fundamental paradigm shift in the way that patterns are identified. Consideration was given to the “importance” of an item in addition to its frequency of occurrence. In this research we propose a novel measure which we term Discriminatory Confidence that identifies the extent to which a given item can segment a dataset in a meaningful manner. We devise an efficient algorithm which is driven by an Information Scoring model that identifies items with high discriminatory power. We compare our results with the classical approach to association rule mining and show that the Information Scoring model produces widely divergent results. Our research reveals that mining on the basis of frequency alone tends to exclude some of the most informative patterns that are discovered using discriminatory power.

Keywords: Association Rule Mining, Discriminatory Confidence, Discriminatory Rules, Information Score Filter.

1 Introduction

Association rules aim to discover patterns of co-occurrences between items in a dataset. Mining potentially useful rules from a given dataset when users have limited knowledge of that particular dataset has attracted considerable interest. Most association rule mining techniques such as Apriori focus on finding sets of items that frequently occur in a dataset and then extract rules based on how a subset of items influence the presence of another subset [1]. Most of the proposed pattern-mining algorithms are variants of the seminal Apriori [2] algorithm.

Apriori and its variants treat every item with equal significance. However, it is sometimes the case that certain items are not informative despite the fact that they occur frequently in a dataset. Consider four items $\{gender = female\}$, $\{age < 20\}$, $\{20 \leq age < 35\}$, and $\{35 \leq age < 50\}$ that occur in a medical (obstetrics) dataset. In the dataset $\{age < 20\}$, $\{20 \leq age < 35\}$, and $\{35 \leq age < 50\}$ are mutually exclusive to each other and occur 30% of the time. Suppose that the minimum support threshold is set at 25%. Suppose also that item $\{gender = female\}$ occurs with 90% probability across each of the three age ranges, giving it a support of occurrence of 0.81, assuming that age and gender are independent of each other. Clearly item $\{gender = female\}$

has very little discriminatory power, despite the fact that it easily meets the given minimum support threshold. This is so not only because it is such a common item, but more importantly because it is incapable of segmenting the dataset into the natural logical partitions represented by $\{age < 20\}$, $\{20 \leq age < 35\}$, and $\{35 \leq age < 50\}$. Thus the extension of item $\{gender = female\}$ with any of the other 3 items do not provide value to the pattern mining process and therefore the unnecessary generation of candidate itemsets such as $\{gender = female, age < 20\}$, $\{gender = female, 20 \leq age < 35\}$, and $\{gender = female, 35 \leq age < 50\}$ can be avoided. Such pruning provides two benefits: increased efficiency and a reduction in the generation of trivial, uninteresting rules that tend to unnecessarily increase the cognitive load on the end-user. Consider a risk factor associated with gynaecological disorders, which is encoded as items $\{risk = moderate\}$, $\{risk = low\}$ and $\{risk = high\}$. If these items co-occur very frequently with items $\{age < 20\}$, $\{20 \leq age < 35\}$, and $\{35 \leq age < 50\}$ respectively then it would have been much more productive to generate the candidate itemsets $\{age < 20, risk = moderate\}$, $\{20 \leq age < 35, risk = low\}$, and $\{35 \leq age < 50, risk = high\}$.

More recent research [2,3,4,5,6,7,8,9] on constraint-based mining has attempted to address this problem. Constraints provide a focus on interesting knowledge, thus limiting the number of patterns extracted to those of potential interest, as defined by the constraint formulae. While such approaches are certainly useful in many situations they do require advance knowledge of both the domain and the distribution of items in the dataset. Such information is not always forthcoming. Even when such information is readily available the knowledge extracted is shaped by the constraints defined which act as pattern templates in some sense. New, unexpected patterns that do not fall within the ambit of these constraints will unfortunately remain hidden.

In this research, we make two novel contributions to the field of association rule mining. Firstly, we propose a novel method of identifying informative items at an early stage in the mining process. Secondly, we devise a novel measure which we term Discriminatory Confidence to evaluate the effectiveness of the set of items extracted. We show formally that a high Information Score is a necessary condition for achieving a high level of discriminative power.

The rest of the paper is organized as follows. In the next section we review related work done in the area of set closure and constraint-based association rule mining. In Section 3 we formally define the problem that is being investigated and present a definition for the discriminatory confidence measure. Section 4 describes the information score filter and establishes a strong link between the concepts of discriminatory confidence and information score. Our experimental results are presented in Section 6. Finally we summarize our research contributions in Section 7 and outline directions for future work.

2 Related Work

In this section we review research on constraint-based mining, an area that is most closely related to our goal of generating informative rule bases. Constraint-based mining enables users to provide restraints to search for useful patterns. Constraints can be categorized into several classes according to their interaction with the mining process.

Succinct constraints are pushed into the initial data selection process at the start of mining, anti-monotonic constraints are pushed deep to restrain pattern growth during mining, while monotonic constraints are used to ensure that subsequent superset patterns can be guaranteed to satisfy the same constraints on patterns that are currently under consideration.

Item constraints were first discussed by Srikant et al. [2] who considered the problem of integrating constraints that were boolean expressions over groups of items prior to rule generation. The constraints restrict the items or combinations of items that are allowed to participate in mined rules. Ng et al. [3] introduced two categories of itemset constraints: anti-monotonicity and succinctness, and proposed an algorithm called CAP for handling constraints belonging to these classes within the Apriori framework. Pei and Han [10] developed a new constraint-based frequent itemset discovery method, called CFG, which pushed constraints into the FP-growth method.

In Grahne et al. [4], the third category of constraint, monotonicity, was introduced in the context of mining correlated sets. They extended Brin et al. [5] principle of finding (minimal) correlated (or dependent) sets of objects from large databases. They base their definition of correlation on the chi-squared metric which is widely used by statisticians for testing independence. Pei et al. [6], introduced convertible constraints and methods which enabled these classes of constraints to be pushed deep inside the FP-growth algorithm. Following this, DualMiner [7] algorithm used both monotone and anti-monotone constraints to prune the search space. Kifer et al. [8] presented a method to mine itemsets with restrictions on their variance. Gade et al. [9] introduced a new class of block constraints that determined the significance of an itemset pattern.

The constraint based mining approach was designed to improve the level of user engagement with the mining process and not specifically to improve information content. It has the potential to remove items that are of interest to the end-user when such items violate constraints that were specified on the basis of out-of-date information arising out of changes to the underlying data distribution. Valuable items can also be pruned by error simply due to the unavailability of a sufficient level of domain knowledge.

3 Problem Definition

The frequent pattern mining problem was first introduced for mining association rules between sets of items [1]. The following is a formal statement of association rule mining for a transaction database. Let $I = \{i_1, i_2, \dots, i_m\}$ be the universe of items. A set $X \subseteq I$ of items is called an itemset or pattern. In particular, an itemset with k items is called a k -itemset. Every transaction contains a unique transaction ID tid . A transaction $t = (tid, X)$ is a tuple where X is an itemset. A transaction $t = (tid, X)$ is said to contain itemset Y if $Y \subseteq X$. Let $\{t_1, t_2, \dots, t_n\}$ be the set of all possible transactions called T . A transaction database D is a set of transactions, such that $D \subseteq T$. In effect, D is really a multiset of itemsets.

The *count* of an itemset X in D , denoted by $\text{count}(X, D)$, is the number of transactions in D containing X . The *support* of an itemset X in D , denoted by $\text{supp}(X, D)$, is the proportion of transactions in D that contain X , i.e.,

$$\text{supp}(X, D) = \frac{\text{count}(X, D)}{|D|}$$

where $|D|$ is the total number of transactions in D . Given a user-specified minimum support threshold $minsup \in [0, 1]$, X is called a **frequent itemset** or **frequent pattern** in D if $sup(X, D) \geq minsup$. The *confidence* of a rule $X \rightarrow Y$, is the proportion of transactions in D that contain both XY divided by X , i.e.,

$$conf(X \rightarrow Y, D) = \frac{count(XY, D)}{count(X, D)}$$

The problem with classical Apriori and Apriori-like algorithms is their excessive reliance on the use of support as a filter for item pruning. The support threshold allows every item which fulfils the threshold to survive, regardless of whether the item is informative or not. To address this problem we include a pre-processing item filter that prunes items that have low discriminatory power. We define formally the notion of *value set and discriminatory confidence*.

Definition 1. The **value set** $V(j)$ for a given item j is given by $V(j) = \{v_1, v_2, \dots, v_n\}$ where each $v_j \in V(j)$ is a value expressed in the dataset D for item j . The values v_j can be of type nominal or integer. □

For example, in a dataset an item *Gender* can have a value set of size two, $V(Gender) = \{female, male\}$.

Definition 2. The **neighborhood** $N_{j=k}$ for a given item $j = k$ is the set of items that co-occur with item $j = k$ taken over all instances of the dataset. □

Definition 3. The **discriminatory confidence** $dc(k = v)$ for item $k = v$ is defined by:

$$dc(k = v) = \sum_{j \in N_{k=v}} \max_{i \in V(k)} \left\{ \sum_{v' \in V(j) \setminus i} (conf(k = v \rightarrow j = v')) - (conf(k = v \rightarrow j \in V(j) \setminus v')) \right\}$$

□

For the special case of a dataset containing only binary valued attributes, this definition simplifies to: $dc(k) = \sum_{j \in N_k} (|conf(k = v \rightarrow j = 0) - conf(k = v \rightarrow j = 1)|)$

Essentially, discriminatory confidence as defined above weighs for each item j in k 's neighborhood (denoted by N_k) the viability of segmenting the dataset D for each item j into m logical partitions s_1, s_2, \dots, s_m . For each partition s_i , the difference in confidence of $conf(k = v \rightarrow j \in s_i) - conf(k = v \rightarrow j \notin s_i)$ is evaluated. It does this for every possible partition and notes the partition where the difference is highest. It then repeats this process for every item j in k 's neighborhood, each time accumulating the maximum difference in confidence that results on the optimal partition for the item j that is currently under consideration. We use the maximum function in Definition 3, so that it applicable to both binary and non-binary datasets.

We illustrate the above definition of Discriminatory Confidence with the help of a simplified example taken from the Zoo dataset from the UCI Machine Learning repository. We compute the Discriminatory Confidence of the item *feathers=1* with the help of Table 1 that displays the rule confidence for every possible pair of items that make up our simplified version of the Zoo dataset.

Table 1. Rule Confidence Matrix for Zoo Dataset

Item	eggs = 0	eggs = 1	feathers = 0	feathers = 1	type = 1	type = 2	type = 3
eggs = 0	1.0	0.0	1.0	0.0	1.0	0.0	0.0
eggs = 1	0.0	1.0	0.7	0.3	0.0	0.3	0.1
feathers = 0	0.5	0.5	1.0	0.0	0.5	0.0	0.1
feathers = 1	0.0	1.0	0.0	1.0	0.0	1.0	0.0
type = 1	1.0	0.0	1.0	0.0	1.0	0.0	0.0
type = 2	0.0	1.0	0.0	1.0	0.0	1.0	0.0
type = 3	0.2	0.8	1.0	0.0	0.0	0.0	1.0

In the interests of clarity we assume that the dataset contains only items $feathers=0$, $feathers=1$, $eggs=0$, $eggs=1$, $type=1$, $type=2$ and $type=3$. Note that all item names are abbreviated by their first letter and value in the example that follows. In the evaluation that follows we only compute the difference in confidence between pairs of rules R_1, R_2 where the left rule R_1 references items that co-exist with the item being evaluated (which happens to be $feathers=1$ in this case). Thus items that do not co-exist with $feathers=1$ such as $t=3$ or $t=1$ never appear in the consequent of the left rule R_1 in the rule pair. From definition 2 we have:

$$\begin{aligned}
 dc(f = 1) &= (conf(f = 1 \rightarrow eggs = 1) - conf(f = 1 \rightarrow eggs = 0)) \\
 &\quad + (conf(f = 1 \rightarrow t = 2) - conf(f = 1 \rightarrow t = 3 | t = 1)) \\
 &= (1.0 - 0) + (1.0 - 0) \\
 &= 2
 \end{aligned}$$

We thus observe that $feathers=1$ item has positive discriminatory power of 2. This is because it is able to segment the dataset cleanly on two partitions which are given by: ($eggs=1$ versus $eggs=0$) and ($type=2$ versus $type=1$ or $type=3$). For a given dataset with N attributes the maximum possible discriminatory confidence attainable is $N - 1$. This is due to the fact that for any given item that is drawn from a particular attribute, say (A_i), an item such as ($A_i = v_i$) can discriminate on at most $N - 1$ item pairs: ($A_1 = v_1, A_1 \neq v_1$), ($A_2 = v_2, A_2 \neq v_2$), ..., ($A_{(i-1)} = v_{(i-1)}, A_{(i-1)} \neq v_{(i-1)}$), ($A_{(i+1)} = v_{(i+1)}, A_{(i+1)} \neq v_{(i+1)}$), ..., ($A_N = v_N, A_N \neq v_N$). This means that the item $feathers=1$ has attained the maximum possible discriminatory confidence of 2 for the dataset under consideration.

Items that have high discriminatory confidence are potentially interesting. For example, the item $feathers=1$ has 100% confidence on the rule: $f = 1 \rightarrow t = 2$ while having 0% confidence on the rule $f = 0 \rightarrow t = 2$. This means that the rule $f = 1 \rightarrow t = 2$ defined by the item $feathers=1$ is strong as the existence of $t = 2$ is solely determined by the item $feathers=1$ and nothing else.

In the next section we will discuss a filter that will efficiently sift through the items in any given dataset and return those items with the highest discriminative power.

4 The Design of the Information Score Filter

The design of our filter was inspired by the ‘‘Valency model’’ that was proposed to automatically infer item weights for a weighted association rule miner [11]. The Valency model was shown to be effective in identifying the relative importance of items

in a given dataset based on the patterns of interactions between the items. The Valency model is based on the intuitive notion that an item should be weighted based on the strength of its connections to other items as well as the number of items that it is connected with. Two items that co-occur often relative to each of their individual support values are said to have a high degree of connectivity and are thus weighted higher. Formally, the total connectivity of a given item k is defined by:

$$c_k = \sum_i^n c(ki)$$

where $c(ki)$ is the connectivity between item k and item i and is given by

$$c_{ki} = \frac{\text{count}(ki)}{\text{count}(k)}$$

From the definition of c_{ki} , we see that it measures the conditional probability of seeing item i given item k , and thus c_{ki} is numerically equal to the confidence of the rule: $k \rightarrow i$.

The Valency model also takes into consideration the number of items that a given item occurs with when taken across the entire dataset. Items that occur with a small number of other items are given a higher purity score. The formal definition of purity is given by:

$$p_k = 1 - \frac{\log_2(|I_k|) + \log_2(|I_k|)^2}{\log_2(|U|)^3}$$

where $|U|$ represents the number of unique items in the dataset and $|I_k|$ represents the number of unique items which co-occur with item k . Thus an item that occurs with just one other item attains the maximum purity value of 1. The purity value converges to the minimum value of 0 as the number of linkages increases and become close to the number of items in the universal set of items. A non-linear logarithmic function was used to ensure a sharp drop in purity with the number of linkages. The $\log(|U|)^3$ term in the denominator ensures that the rate of decrease in purity is sensitive to the size of the universal set. For databases with a larger number of items (larger $|U|$) the gradient of descent is steeper when compared to databases with a smaller pool of items (smaller $|U|$) and so a smaller number of items will acquire high purity values.

The information score filter that we developed uses elements of both connectivity and purity. However we modify the manner in which they are combined. We believe that the Information Score for an item should reflect the strength of its connections to its neighboring items. At the same time we expect the purity of an item to play a significant role as purity essentially determines how distinctive a given item is. Accordingly, we define the Information Score for an item k , denoted by i_{S_k} as a multiplicative function of each item's purity and its connectivity with each of its neighbors. The Information Score i_{S_k} is given by:

$$i_{S_k} = \sum_i^n c(ki)p_i \quad (1)$$

The information score as defined above attaches importance to items that are distinctive. When such distinctive items co-occur together strongly with other distinctive items in

small cliques, a high score results for each of the items in the clique. We implement the Information Score filter by building and maintaining the item neighborhood on the fly as the dataset is being scanned. At the end of the scan the information score for each item is computed and a minimum scoring threshold (*minscore*) is used in order to return items with score greater than or equal to the *minscore* value.

We use the *is* filter as a pre-pruning mechanism to identify candidate items with high discriminatory power. The items that survive this filter are then combined together to form candidate itemsets in the usual manner as Apriori. In the next section we show that high information score leads to a high discriminatory power.

5 Discriminatory Confidence vs. Information Score

In this section we formally prove the relationship between Information Score and Discriminatory Confidence.

Theorem 1. *The Discriminative Confidence dc_k for a given item k is determined by its Information Score is_k and is given by: $dc_k > is_k - 2s$, where s is the size of item k 's neighborhood.*

Proof. Consider N_k as the set of items in item k 's neighborhood and V_j as the value set for item j . From Equation 1 we have:

$$is_k = \sum_{j \in N_k} \sum_{l \in V_j} c_{kl} \cdot p_l = \sum_j \in N_k \frac{1}{|V_j|} \left(\sum_{l \in V_j} (c_{kl} p_l + \sum_{m \in V_j \setminus \{l\}} c_{km} p_m) \right)$$

since for any given element $l \in V_j$ the sum of the rest of terms taken over the complement of l (which is m) is equal to the sum over the entirety of l

$$\begin{aligned} &= \sum_{j \in N_k} \frac{1}{|V_j|} \left(\sum_{l \in V_j} c_{kl} p_l - \sum_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} p_m + 2 \sum_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} p_m \right) \\ &< \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} (c_{kl} \cdot p_l - c_{km} \cdot p_m) + 2 \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} p_m \\ &< p_{\max} \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} (c_{kl} - c_{km}) + 2p_{\max} \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} \end{aligned}$$

where p_{\max} is the maximum purity over items in items k 's neighborhood. However from the definition of purity, $p_{\max} \leq 1$, since the purity of any given item takes a maximum value of 1. Thus,

$$is_k < \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} (c_{kl} - c_{km}) + 2 \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km}$$

Now from Definition 3 for discriminatory confidence we have:

$$\sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} (c_{kl} - c_{km}) = dc(k)$$

given the numerical equivalence of connectivity between any given pair of items and the confidence of the rule that they define. So,

$$is_k < dc_k + 2 \sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km}$$

Now $\forall l \in V_j, \sum c_{km} \leq 1$ since $m \subset V_j$ and $\sum_{l \in V_j} c_l = 1 \forall l \in V_j$. We thus have:

$$\max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} \leq 1$$

and so

$$\sum_{j \in N_k} \max_{l \in V_j} \sum_{m \in V_j \setminus \{l\}} c_{km} \leq s$$

giving:

$$dc_k > is_k - 2s$$

This result shows that for a given size of item ($k = v$)’s neighborhood (of size s), the discriminatory confidence (dc_k) increases with increasing information score (is_k). As the items I_1, I_2, \dots, I_s have greater purity and connectivity with item $k = v$, the information score for item ($k = v$) increases, which in turn causes a corresponding increase in dc_k .

The ultimate goal in identifying discriminatory items is to produce stronger rules having discriminatory items in their rule terms. Before presenting a formal definition of discriminatory rules we discuss a few motivating examples. In the Zoo dataset, the two items *feathers=1* and *type=2* have high discriminative power on their own. Furthermore, they also discriminate between each other. That is, when the item *feathers=1* occurs, the other item *type=2* occurs as well. Thus a rule such as *feathers=1* \rightarrow *type=2* expresses the fact that with a high level of confidence, which happens to be 1 in this case, that the item *type=2* is associated with *feathers=1* and *not feathers=0*. Such types of rules are valuable as they capture in the most concise manner possible the trigger conditions for the rule consequent to be true, while maintaining the highest level of confidence.

In the general case, given a rule ($X = v_1 \rightarrow Y = v_2$), involving two items $X = v_1$ and $Y = v_2$, the greater the discriminative power of $X = v_1$ with respect to $Y = v_2$, the lesser is the need for the item $X = v_1$ to be augmented by other items in the antecedent part of the rule. In the above example whenever we have the occurrence of *feathers=1* there will be no need to check the status of other items in order to deduce the type of the animal. In a medical diagnosis scenario, a rule linking a symptom (or two) with very high discriminatory power with respect to a certain disease could be sufficient for a physician to make an initial prognosis which can be later refined into a diagnosis through the use of confirmatory tests. Likewise, a discriminatory rule defined on a web click stream environment involving a short sequence of clicks which are discriminatory with respect to a subsequent destination can be used to recommend that destination to the user, with a high level of confidence that the user will find that destination to be of use.

With the above in mind we now present a formal definition of discriminatory rules. In the discussion that follows we use the term “item” to stand for either a singleton item or a set of items (i.e. an itemset).

Taking the above into consideration we define a discriminatory rule in the following manner.

Definition 4. *The rule $X = v_1 \rightarrow Y = v_2$ is said to be a **discriminatory rule** if:*

1. $dc(X = v_1) \geq 0$ and $dc(Y = v_2) \geq 0$.
2. $conf(X = v_1 \rightarrow Y = v_2) > \alpha$ where α is a user specified threshold on rule confidence.
3. Item $X = v_1$ must discriminate on item $Y = v_2$, that is $conf(X = v_1 \rightarrow Y = v_2) - conf(X \neq v_1 \rightarrow Y = v_2) * supp(X = v_1, Y = v_2) > rdf$ where rdf is another user specified threshold called the Rule Discriminatory Factor that defines the discriminative power of the rule. \square

The discriminative power of $X = v_1$ with respect to $Y = v_2$ is measured in part by:

$$F = conf(X = v_1 \rightarrow Y = v_2) - conf(X \neq v_1 \rightarrow Y = v_2)$$

However, the above difference in confidence alone does not express the true discriminative power of a rule, as rules with a larger number of terms in their antecedent portion would tend to acquire a higher value for F , at the expense of rule support which would be lower due to the higher number of antecedent terms. To compensate for this, we multiple F by the rule support, $supp(X = v_1, Y = v_2)$.

One possible method of generating discriminatory rules would be to scan the dataset and to identify items with positive discriminatory power. This direct approach, although appealing due to its conceptual simplicity, is unfortunately computationally expensive. This approach would require a scan of the dataset to first build item neighborhoods. Once such neighborhoods are built in the form of a connectivity graph a further scan of the connectivity graph would be required in order to evaluate the discriminatory confidence measure for each item. This second scan will involve, for each item k , the computation of the union of sets of items that occur with item k , as shown in the example for the computation of dc value for the item *feathers=1*. The greater the number of non binary valued attributes the greater is the number of union operations that need to be performed. For each n -ary attribute $n(n - 1)$ union operations are required.

As such we generate rules by computing the information score for each item and then use a threshold, *minscore*, to filter out non informative items. All singleton items are extended into itemsets in exactly the same manner as Apriori combines frequent items together, except that the criterion for itemset extension is not support but information score. For an item $X = v_1$ to be extended with item $Y = v_2$, $\frac{is(X=v_1)+is(Y=v_2)}{2}$ must be \geq *minscore*. Thus the difference between our rule generator and classical Apriori is the criterion that is applied for item generation and extension. As our experimental results in the next section will show the use of the information score model not only produces items with greater discriminative power than classical rule generators, but also produces rules that can be put to better use in practice.

6 Empirical Study

Our motivation in introducing the information score model was to automatically sift non-informative items from informative ones without requiring additional information from users. As such we were interested in examining the impact of the filter in an environment where such user input is not available. This led us to compare our algorithm with the classical Apriori approach. Our experimentation was conducted in three steps; firstly, a performance comparison of information score measure against support was conducted. Secondly, we examined the correlation of rules produced by *is* filter vis-a-vis Apriori, and lastly we assessed the discriminative power of rules generated through the use of the *is* filter.

Table 2. Results on Zoo Datasets

Information Score Filter		Support			
Item	dc value	count	Item	dc value	count
feathers = 1	11.7	20	venomous = 0	-8.02	93
type = 2	11.7	20	domestic = 0	-8.02	88
type = 6	11.8	8	fins = 0	-7.02	84
type = 4	12.8	13	backbone = 1	-3.10	83
legs = 6	8.8	10	feathers = 0	-7.02	81
legs = 2	3.9	27	breathes = 1	-6.03	80
fins = 1	5.9	17	airbourne = 0	-8.03	77
airbourne = 1	1.8	24	tail = 1	-5.03	75
milk = 1	5.0	41	aquatic = 0	-4.03	65
type = 1	5.0	41	toothed = 1	-0.07	61

6.1 Discriminatory Confidence Analysis

In this section, we experiment with seven datasets taken from the UCI machine learning repository [12]. We evaluate our results with the help of the *dc* measure that we developed. In each case we select the top 10 items for each algorithm based on their respective ranking criteria which happens to be Information Score for our approach and the Support measure for Apriori. The *dc* measure is only used as an evaluation mechanism and is not a part of the *is* filter implementation.

Table 2 shows the Top 10 items ranked by the *is* score and support. The top ten items ranked by *is* all have a positive *dc* value, whereas this is not the case for items ranked by the support measure. Consider the item “*feathers = 1*” which happens to be the item ranked highest by the *is* filter. This item occurs 20 times in the dataset. In stark contrast, “*venomous = 0*” which is Apriori’s top performer has no discriminatory power and occurs as frequently as 93 times in a dataset that contains 101 data instances. It also happens to occur with every other item. Clearly this is a prime candidate for pruning if the mining objective is to produce rules that have a high degree of informative content. The item “*domestic = 0*” follows a similar trend as well. When we correlated the rankings of items produced by the *is* score with the *dc* measure we obtained an 83% correlation, thus supporting the assertion in Theorem 1.

Table 3. Results on UCI Datasets

Dataset	is filter		Support		
	Item	dc	Item	dc	
Mushroom	odor = m	16.3	veil-type = p	-72.6	
	classes = cyst-nematode	7.7	veil-color = w	-62.6	
	stalk-color-above-ring = c	16.3	gill-attachment = f	-63.6	
	stalk-color-below-ring = c	16.3	ring-number = o	-60.5	
	ring-number = n	16.3	gill-spacing = c	-65.6	
	ring-type = n	16.3	gill-size = b	-56.2	
	spore-print-color = o	14.5	stalk-surface-above-ring = s	-57.6	
	spore-print-color = y	14.5	stalk-surface-below-ring = s	-52.6	
	spore-print-color = b	14.5	bruises = f	-56.0	
	gill-color = o	13.5	stalk-shape = t	-22.8	
	veil-color = n	12.5	mycelium = 0	-44.1	
	Soybean	fruit-pods = 2	7.7	sclerotia = 0	-42.2
		classes = herbicide-injury	9.5	leaf-malf = 0	-40.1
classes = diaporthe-pod-stem-blight		16.0	roots = 0	-38.0	
classes = charcoal-rot		22.8	leaf-mild = 0	-37.1	
int-discolor = 2		22.8	shriveling = 0	-39.0	
sclerotia = 1		22.8	seed-size = 0	-39.0	
roots = 2		-14.4	lodging = 0	-39.0	
classes = rhizoctonia-root-rot		22.8	mold-growth = 0	-40.2	
classes = downy-mildew		17.8	seed-discolor = 0	-38.0	
Hepatitis		PROTIME = 85	4.50	X-class = 0	-13.01
	region = 1	7.88	SEX = 1	-249.26	
	ALKPHOSPHATE = 81	-3.00	VARICES = 2	-236.82	
	PROTIME = 74	-2.50	ANTIVIRALS = 2	-248.47	
	SGOT = 30	-4.20	ASCITES = 2	-232.75	
	ALBUMIN = 4.5	-1.50	Class = 2	-220.00	
	SGOT = 55	-6.60	ANOREXIA = 2	-223.79	
	SGOT = 18	-1.00	SPLEENPALPABLE = 2	-228.67	
	SGOT = 60	-4.33	LIVERBIG = 2	-239.53	
	BILIRUBIN = 4.6	-4.00	FATIGUE = 1	-220.00	

Table 3 shows the same trend as Table 2 for 3 other real-world datasets, whereby the Top 10 items ranked by information score generally produced positive discriminatory values. In contrast, the Top 10 items ranked by support were never able to generate positive discriminatory confidence values. The results for the other 3 datasets produced similar results but has been omitted due to space constraints. Overall, we have experimentally established that items with high *is* score display high discriminatory power. On average there is a high correlation between ranking of items ranked by *is* score with their *dc* values; this average is 0.93 when taken across all datasets. Some datasets return negative *dc* values for certain items with the *is* filter; this is due to the inherent nature of the dataset. Such datasets contain fewer discriminatory items.

We also recorded the execution times for *is* filter and Apriori. Overall, even with the additional overhead of preprocessing the data the average execution time (taken over all datasets that we experimented with) was 1 second with *is* filter whereas it was 151 seconds for Apriori. Although the *is* filter introduces its own overheads in the pre-pruning phase it compensates by passing fewer items for itemset extension than Apriori, thus causing a significant reduction in the overall execution time.

6.2 Correlation Analysis

Correlation analysis was first introduced by Brin et al. [5]. We used the Chi square χ^2 test to assess the degree of independence between the antecedent and consequent of a rule. In Table 4 each row contains the name of the dataset, followed by the number of itemsets produced, and the percentage of rules generated which are correlated (at a significance threshold of 0.1). We varied the minimum support threshold between 0.1-0.5 for Apriori. For the *is* filter method we used minscore values in the range 0.85 to 0.90 for the pre-pruning phase and thereafter used a minimum support threshold of 0.01 when extending the surviving items into itemsets. We experimentally verified that these

Table 4. Correlation Analysis for UCI Datasets

Dataset	Information Score Filter		Apriori	
	Itemset	% Corr	Itemset	% Corr
Zoo	33	100	11588	99
Flare	372	100	14121	70
Primary Tumor	199	100	15872	37
Flag	161	100	41968	69
Hepatitis	506	100	37187	94
Soybean-Large	339	100	32285	99
Mushroom	9649	100	5477	99

thresholds for *is* filter yielded rules that covered the support range produced by Apriori’s rule base. From Table 4 we notice that the percentage of correlated rules ranges from 37% to 99.0% for Apriori, whereas the Information Score filter achieved a perfect score of 100% in all cases.

6.3 Discriminatory Rule Analysis

In this section we contrast the discriminatory power of the two rule generators. In all of our experimentation we used as metrics: Gini Index (G) which measures the amount of Information Gain for a rule, rule Lift and the percentage of rules that had an RDF value at or above the 80th percentile or the 40th percentile, identified as RDF80 and RDF40 respectively in Table 5. The Gini Index is an interest measure that has been widely applied in pattern mining and measures the amount of information gained by splitting a dataset according to a given criterion [13]. Together with the RDF measure that we defined, it is thus suited to evaluate how successful our Information Score model is able to rank items in terms of their ability to create partitions in datasets that have high information content.

Table 5 clearly shows that the Information Score filter outperforms Apriori on all 3 metrics that we tracked. Improvements in the G metric ranged from 20% for the Hepatitis dataset to 353% for the Flag dataset. Significant improvements in the average Lift value was also observed for Information Score, with the exception of Primary-Tumor where Apriori had a slightly higher Lift average. The discriminatory power for Information Score was much higher at the 80th percentile with Apriori having no more than 32% of its rules being in the discriminative category (with the exception of the Zoo dataset).

We now take a close look at the rules produced by the Zoo dataset. In the case of the Information Score filter, we noticed that discriminatory rules were more concise than with Apriori. We compared the two methods for a given rule consequent. For example,

Table 5. Discriminatory Rule Analysis for UCI Data

Dataset	Apriori					Information Score Filter				
	Rules	Average G index	RDF80	RDF40	Lift	Rules	Average G index	RDF80	RDF40	Lift
Flag	155524	.015	.15	6.3	1.1	4527	.068	98.2	100.0	9.1
Flare	244753	.069	31.7	48.2	2.4	822	.112	48.1	52.0	5.1
Hepatitis	720633	.060	30.0	82.2	1.8	463	.072	63.3	93.5	1.5
Mushroom	61515	.138	32.1	67.8	1.0	747	.168	80.3	83.5	3.4
Primary Tumor	37486	.005	.03	1.6	1.1	33423	.019	22.6	55.2	1.5
Soybean	188223	.140	10.3	85.8	1.2	1165	.238	98.5	100.0	1.8
Zoo	644890	.298	77.0	83.8	2.4	206	.368	96.6	100.0	2.4

hair = 1 milk = 1 →type = 1
hair = 1 →type = 1
hair = 1 legs = 4 →type = 1
legs = 4 milk = 1 →type = 1
hair = 1 legs = 4 milk = 1 →type = 1
hair = 1 legs = 2 →type = 1
legs = 2 milk = 1 →type = 1
hair = 1 legs = 2 milk = 1 →type = 1
breathes = 0 fins = 1 →type = 4
breathes = 0 fins = 1 legs = 0 →type = 4
feathers = 1 →type = 2
feathers = 1 legs = 2 →type = 2
airbourne = 1 feathers = 1 →type = 2
airbourne = 1 feathers = 1 legs = 2 →type = 2

Fig. 1. Extract of rules produced

we took the rule consequent “*type = I*”, and found that were 9 different rules produced by Information Score filter as compared to 4584 rules produced by Apriori for the same consequent. The rule with the highest lift value was $\{\text{milk} = 1\} \rightarrow \{\text{type} = 1\}$ with a lift value of 2.7 which also produced the highest G value of 0.48. Overall, the rules produced by Apriori had on average about 6 terms in the antecedent as opposed to IS which had an average of about 1.8 terms in the antecedent. The other rules with the same consequent produced by IS filter are listed at the top of Figure 1.

In general, the other rules produced were concise and gave precise representations of different partitions within the dataset. For example, most of the items in the antecedent are distinctive items that are able to characterize a particular type of animal in the Zoo dataset. This situation generalizes to all the rule bases produced with the IS filter.

7 Conclusion

This research has revealed that the standard approach of pruning items based on their support value alone is not optimal from the standpoint of generating informative rules. We established a clear nexus between two novel concepts that we introduced: Information Score and Discriminative Confidence. Items with high Information Score were shown to have high discriminatory power. We experimentally established that items that discriminate between each other gives rise to more informative and actionable rules.

In this research we were guided by the twin objectives of improving information content and reducing cognitive load on the end-user of a rule base. These two objectives are actually synonymous with each other, improvement of information content tends to produce a more compact rule base which in turn enables the end-user to exploit the rules more effectively.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)

2. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: KDD, pp. 67–73 (1997)
3. Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. *SIGMOD Rec.* 27(2), 13–24 (1998)
4. Grahne, G., Wang, X., Lakshmanan, L.V.: Efficient mining of constrained correlated sets. In: International Conference on Data Engineering, p. 512 (2000)
5. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. *SIGMOD Records* 26(2), 265–276 (1997)
6. Pei, J., Han, J., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: ICDE 2001: Proceedings of the 17th International Conference on Data Engineering, p. 433. IEEE Computer Society, Washington, DC, USA (2001)
7. Bucila, C., Gehrke, J., Kifer, D., White, W.: Dualminer: a dual-pruning algorithm for itemsets with constraints. In: KDD 2002: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 42–51. ACM (2002)
8. Kifer, D., Gehrke, J., Bucila, C., White, W.: How to quickly find a witness. In: PODS 2003: Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 272–283. ACM, New York (2003)
9. Gade, K., Wang, J., Karypis, G.: Efficient closed pattern mining in the presence of tough block constraints. In: KDD 2004: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 138–147. ACM (2004)
10. Pei, J., Han, J.: Can we push more constraints into frequent pattern mining? In: KDD 2000: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 350–354. ACM, New York (2000)
11. Koh, Y.S., Pears, R., Yeap, W.: Valency Based Weighted Association Rule Mining. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6118, pp. 274–285. Springer, Heidelberg (2010)
12. Asuncion, A., Newman, D.: UCI machine learning repository (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
13. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proc. of the 2002 ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, Alberta, Canada (2002)

Dominance-Based Soft Set Approach in Decision-Making Analysis

Awang Mohd Isa¹, Ahmad Nazari Mohd Rose¹, and Mustafa Mat Deris²

¹Faculty of Informatics

Universiti Sultan Zainal Abidin, Terengganu, Malaysia

²Faculty of Information Technology and Multimedia

Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

{isa, anm}@unisza.edu.my, mmustafa@uthm.edu.my

Abstract. Multi-criteria decision analysis, sometimes called multi-criteria decision making, is a discipline aimed at supporting decision makers faced with making numerous and sometimes conflicting evaluations. Multi-criteria decision analysis aims at highlighting these conflicts and providing a compromised solution in a transparent process. This paper introduces the application of soft-dominance relation based on soft set theory in the field of multi-criteria decision analysis. This relation is an extension of the soft set theory which deals with typical inconsistencies during the consideration of criteria and in preference-ordered decision classes. The paper also utilized soft-dominance relations based on soft set theory in obtaining the decision rules in dealing with problems in a multi-valued information system.

Keywords: Multi-criteria decision analysis, Soft set theory, Multi-soft set, Soft-dominance relation, Multi-valued Information System.

1 Introduction

Many decision-making problems are characterized by the ranking of objects according to a set of criteria with pre-defined preference-ordered decision classes, such as credit approval [18], stock risk estimation [1], mobile phone alternatives estimation [10]. Models and algorithms were proposed for extracting and aggregating preference relations based on distinct criteria. The underlying objectives are to understand the decision process, to build decision models and to learn decisions rules from data.

Rough set theory provides an effective tool for dealing with inconsistency and incomplete information [[19],[14]]. It has been widely applied in feature evaluation [17], attribute reduction and rule extraction [9]. Pawlak's rough set model is constructed based on equivalence relations. These relations are viewed by many to be one of the main limitations when employing the model to complex decision tasks. However, there is an extension of the rough set [6] to deal with these limitations.

In multiple criteria decision-making problems, there are preference structures between conditions and decisions. Greco et al. [6] introduced a dominance rough set

model that is suitable for preference analysis. In [6], the decision-making problem with multiple attributes and multiple criteria were examined, where dominance relations were extracted from multiple criteria and similarity relations were constructed from numerical attributes and equivalence relations were constructed from nominal features. An extensive review of multi-criteria decision analysis based on dominance rough sets is given in [7]. Dominance rough sets have also been applied to ordinal attribute reduction and multi-criteria classification. While the theory rough set is well-known and often useful approach to describe uncertainty, it has inherent difficulties as pointed by Molodtsov [13].

Soft set theory proposed by Molodtsov [13] provides an effective tool for dealing with inconsistencies which is free from the difficulties affecting existing methods. As reported in [13], a wide range of applications of soft sets have been developed in many different fields. There has been a rapid growth of interest in soft set theory and its application especially in decision making in recent years. Maji et al. [12] discussed the application of soft set theory in a decision making. Based on fuzzy soft sets, Roy and Maji [15] presented a method of object recognition from an imprecise multi-observer data and applied it to decision making problems. Chaudhuri et al [2] define the concepts of Soft Relation and Fuzzy Soft Relation and then apply them to solve a number of Decision Making Problems. Feng et al [3] introduce an adjustable approach to fuzzy soft set and investigate the application of the weighted fuzzy soft set in decision making. Feng et al [4] also present the application of level soft sets in decision making based on interval-valued fuzzy soft sets. Jiang et al [11] present an adjustable approach to intuitionistic fuzzy soft sets based decision making by using level soft sets of intuitionistic fuzzy soft sets.

Molodtsov's proposal of soft set has been studied and applied by several authors in the cases of uncertainty in decision making. Currently, there are not much literature discussing the application of soft set in the area of multicriteria decision making dealing with uncertainty. Thus, our paper intends to study the feasibility of applying dominance relation based on soft set theory in situations shrouded by uncertainty during the process of multicriteria decision making. In the proposed scheme, the decision system is transformed into the equivalent multi-soft set where in each soft set, the predicates are ordered according to the preference order, and then the approximations will be obtained using dominance-based soft set approach (DSSA), that is an extension of the soft set theory.

The rest of this paper is organized as follows. Section 2 describes the fundamental concept of information systems. In section 3, we present the concept of soft set theory for multi-valued information systems. The extension of soft set approach based on dominance principle is introduced in section 4. An illustrative example is given in section 5 followed by the conclusion of our work is described in section 6.

2 Information System

An information system is a 4-tuple (quadruple), $S = (U, A, V, f)$, where U is a non-empty finite set of objects, A is a non-empty finite set of attributes, $V = \bigcup_{a \in A} V_a$, V_a

is the domain (value set) of attribute a , $f : U \times A \rightarrow V$ is a total function such that $f(u, a) \in V_a$, for every $(u, a) \in U \times A$, called information (knowledge) function. An information system is also called a knowledge representation systems or an attribute-valued system that can be intuitively expressed in terms of an information table (as shown in Table 1).

Table 1. An information system

U	a_1	a_2	...	a_k	...	$a_{ A }$
u_1	$f(u_1, a_1)$	$f(u_1, a_2)$...	$f(u_1, a_k)$...	$f(u_1, a_{ A })$
u_2	$f(u_2, a_1)$	$f(u_2, a_2)$...	$f(u_2, a_k)$...	$f(u_2, a_{ A })$
u_3	$f(u_3, a_1)$	$f(u_3, a_2)$...	$f(u_3, a_k)$...	$f(u_3, a_{ A })$
\vdots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
$u_{ U }$	$f(u_{ U }, a_1)$	$f(u_{ U }, a_2)$...	$f(u_{ U }, a_k)$...	$f(u_{ U }, a_{ A })$

In many applications, there is an outcome of classification that is known. This knowledge, which is known as *a posteriori* knowledge is expressed by one (or more) distinguished attribute called decision attribute. This process is known as supervised learning. An information system of this kind is called a decision system. A decision system is an information system of the form $D = (U, A \cup \{d\}, V, f)$, where $d \notin A$ is the decision attribute. The elements of A are called condition attributes. Condition attributes with value sets are ordered according to decreasing or increasing preference of a decision maker are called criteria.

3 Soft Set Theory

Throughout this section U refers to an initial universe, E is a set of parameters, $P(U)$ is the power set of U and $A \subseteq E$.

Definition 1. (See [13].) A pair (F, A) is called a soft set over U , where F is a mapping given by

$$F : A \rightarrow P(U).$$

In other words, a soft set over U is a parameterized family of subsets of the universe U . For $\varepsilon \in A$, $F(\varepsilon)$ may be considered as a set of ε -elements of the soft set (F, A) or as the set of ε -approximate elements of the soft set. Clearly, a soft set is not a (crisp) set. As for illustration, Molodtsov has considered several examples in [13]. The example shows that, soft set (F, A) can be viewed as a collection of approximations, where each approximation has two parts:-

- (i) A predicate p ; and
- (ii) An approximate value-set v (or simply to be called value-set v).

We denote $(F, A) = \{p_1 = v_1, p_2 = v_2, \dots, p_n = v_n\}$, where n is a number of predicates.

Based on the definition of an information system and soft set, we then show that a soft set is a special type of information systems, i.e., a binary-valued information system.

Proposition 2. (See [8].) *If (F, A) is a soft set over the universe U , then (F, A) is a binary-valued information system $S = (U, A, V_{\{0,1\}}, f)$.*

Proof. Let (F, A) be a soft set over the universe U , we define a mapping

$$F = \{f_1, f_2, \dots, f_n\},$$

where

$$f_i : U \rightarrow V_i \text{ and } f_i(x) = \begin{cases} 1, & x \in F(a_i) \\ 0, & x \notin F(a_i) \end{cases}, \text{ for } 1 \leq i \leq |A|.$$

Hence, if $V = \bigcup_{a_i \in A} V_{a_i}$, where $V_{a_i} = \{0,1\}$, then a soft set (F, A) can be considered as a binary-valued information system $S = (U, A, V_{\{0,1\}}, f)$.

From Proposition 2, it can be easily understood that a binary-valued information system can be represented using soft set theory. Thus, we can make a one-to-one correspondence between (F, E) over U and $S = (U, A, V_{\{0,1\}}, f)$.

Definition 3. (See [12].) *The class of all value sets of a soft set (F, E) is called value-class of the soft set and is denoted by $C_{(F,E)}$.*

Proposition 4. (See [8].) *If $(F, A) = \{(F, a_i) : 1 \leq i \leq |A|\}$ is a multi-soft set over the universe U , then (F, A) is a multi-valued information system $S = (U, A, V, f)$.*

Proof. Let $S = (U, A, V, f)$ be a multi-valued information system and $S^i = (U, a_i, V_{a_i}, f)$, $1 \leq i \leq |A|$ be the $|A|$ binary-valued information systems. From Proposition 2, we have

$$\begin{aligned} S = (U, A, V, f) &= \begin{cases} S^1 = (U, a_1, V_{\{0,1\}}, f) & \Leftrightarrow (F, a_1) \\ S^2 = (U, a_2, V_{\{0,1\}}, f) & \Leftrightarrow (F, a_2) \\ \vdots & \vdots \\ S^{|A|} = (U, a_{|A|}, V_{\{0,1\}}, f) & \Leftrightarrow (F, a_{|A|}) \end{cases} \\ &= ((F, a_1), (F, a_2), \dots, (F, a_{|A|})) \\ &= \{(F, a_i) : 1 \leq i \leq |A|\} \end{aligned}$$

It is proved that $(F, A) = \{(F, a_i) : 1 \leq i \leq |A|\}$ is a *multi-soft sets* over universe U representing a multi-valued information system $S = (U, A, V, f)$.

Since the definition of soft sets is based on the mapping of value sets to the set of objects, it can then only handle one kind of inconsistency of decision - the one related to the inclusion of objects into different value-set of attributes, i.e., predicates. While this is sufficient for classification of taxonomy type, the classical soft set approach fails in case of ordinal classification with monotonicity constraints [16], where the value-sets of attributes are preference ordered. In this case, decision examples may be inconsistent in the sense of violation of the dominance principle which requires that an object x dominating object y on all considered criteria (i.e., x having evaluations at least as good as y on all considered criteria).

4 Dominance Relation Based Soft Set Theory

In this section, we present the principle of dominance relation based on soft set theory or dominance-based soft set approach (DSSA), that can be used in decision-making analysis. As mention the previous section, multi-valued information system $S = (U, A, V, f)$ can be represented by a multi-soft set $(F, A) = \{(F, a_i) : 1 \leq i \leq |A|\}$, where A is a finite set of parameters representing the set of attributes in multi-valued information system. The set A is, in general, divided into set C of condition attributes and set D of decision attributes.

Condition attributes with value sets ordered according to decreasing or increasing preference of a decision maker are called criteria. Since for each criterion $c \in A$ is represented by criterion soft set $C = (F, c)$ in multi-soft set (F, A) , the value sets for criterion c is equivalence to the value sets of the soft set when the predicates are ordered according to decreasing or increasing preference. For soft set $C = (F, c)$, \succeq_c is an outranking relation on U with reference to soft set $C \in (F, A)$ such that $x \succeq_c y$ means “ x is at least as good as y with respect to soft set C ”.

Definition 5. For criterion soft set $C = \{p_i = v_i, i = 1, 2, \dots, n\}$, for all $r, s \in \{1, \dots, n\}$, such that $r > s$, predicate p_r is dominance than predicate p_s with respect to C , if the value of p_r is preferred to p_s , and we denote that by $p_r \succeq_c p_s$.

Definition 6. For criterion soft set $C = \{p_i = v_i, i = 1, 2, \dots, n\}$, the set of objects in value-set v_r are preferred to the set of objects in value-set v_s with respect to C , if predicate p_r is dominance than predicate p_s , we denote it by $v_r \succeq_c v_s$ iff $p_r \succeq_c p_s, \forall r, s \in \{1, \dots, n\}$.

Furthermore, let suppose that the set of decision attributes D is a singleton $\{d\}$ and is represented by decision soft set $D = (F, d)$. The values of predicate in D make a partition of universe U into a finite number of decision classes, $Cl = \{Cl_t, t = 1, \dots, n\}$,

such that each $x \in U$ belongs to one and only one class $Cl_t \in Cl$. It is supposed that the classes are preference-ordered, i.e. for all $r, s \in \{1, \dots, n\}$, such that $r > s$, the objects from Cl_r are preferred to objects from Cl_s . If \succeq is a comprehensive weak preference relation on U , i.e. if for all $x, y \in U$, $x \succeq y$ means “ x is comprehensively at least as good as y ”, it is supposed: $[x \in Cl_r, y \in Cl_s, r > s] \Rightarrow [x \succeq y \text{ and not } y \succeq x]$. The above assumptions are typical for consideration of ordinal classification problems (also called multiple criteria sorting problems).

The set to be approximated are called *upward union* and *downward union* of classes, respectively:

$$Cl_t^{\succeq} = \bigcup_{s \geq t} Cl_s, Cl_t^{\preceq} = \bigcup_{s \leq t} Cl_s, t = 1, \dots, n.$$

The statement $x \in Cl_t^{\succeq}$ means “ x belongs to at least class Cl_t ”, while $x \in Cl_t^{\preceq}$ means “ x belongs to at most class Cl_t ”. Let us remark that $Cl_1^{\succeq} = Cl_n^{\preceq} = U$, $Cl_n^{\succeq} = Cl_1^{\preceq}$ and $Cl_t^{\preceq} = Cl_{t-1}$. Furthermore, for $t = 2, \dots, n$,

$$Cl_{t-1}^{\preceq} = U - Cl_t^{\succeq} \text{ and } Cl_t^{\succeq} = U - Cl_{t-1}^{\preceq}.$$

The key idea of the soft set approach is representation (approximation) of knowledge generated by decision soft set, using granules of knowledge generated by criterion soft sets. In DSSA, the knowledge to be represented is a collection of upward and downward unions of classes, and the granules of knowledge are sets of objects defined using a soft dominance relation.

x dominates y with respect to $P \subseteq C$ (shortly, $x P_{soft}$ -dominates y), denoted by $x D_p y$, if for every criterion soft set $q \in P$ and $x \in v_i, y \in v_j: v_i \succeq_q v_j$. The relation P_{soft} -dominance is reflexive and transitive, i.e. it is a partial order.

Given a set of soft sets $P \subseteq C$ and $x \in U$, the “granules of knowledge” used for approximation in DSSA are:

- a set of objects dominating x , called P_{soft} -dominating set,

$$D_p^+(x) = \{y \in U : y D_p x\},$$

- a set of objects dominated by x , called P_{soft} -dominated set,

$$D_p^-(x) = \{y \in U : x D_p y\}.$$

From the above, it can be seen that the “granules of knowledge” have the form of upward (positive) and downward (negative) dominance cones in the evaluation space. Recall that the dominance principle requires that an object x dominating object y on all considered criterion soft sets or criteria (i.e., x having evaluation at least as good as y on all considered soft set) should also dominate y on decision soft set (i.e., x should be assigned to at least as good decision class as y). This is the only principle widely agreed upon in the multiple criteria comparisons of objects.

Given $P \subseteq C$, the inclusion of an object $x \in U$ to the upward union of classes Cl_t^{\geq} ($t = 2, \dots, n$) is inconsistent with the dominance principle if one of the following conditions holds:

- x belongs to class Cl_t or better, but it is P_{soft} -dominated by an object y belonging to a class worse than Cl_t , i.e. $x \in Cl_t^{\geq}$ but $D_p^+(x) \cap Cl_{t-1}^{\leq} \neq \emptyset$,
- x belongs to a worse class than Cl_t but it P_{soft} -dominates an object y belonging to class Cl_t or better, i.e. $x \notin Cl_t^{\geq}$ but $D_p^-(x) \cap Cl_{t-1}^{\geq} \neq \emptyset$.

If, given a set of soft set $P \subseteq C$, the inclusion of $x \in U$ to Cl_t^{\geq} ($t = 2, \dots, n$) is inconsistent with the dominance principle, then x belongs to Cl_t^{\geq} with some ambiguity. Thus, x belongs to Cl_t^{\geq} without any ambiguity with respect to $P \subseteq C$, if $x \in Cl_t^{\geq}$ and there is no inconsistency with the dominance principle. This means that all objects P_{soft} -dominating x belong to Cl_t^{\geq} , i.e., $D_p^+(x) \subseteq Cl_t^{\geq}$.

Furthermore, x possibly belongs to Cl_t^{\geq} with respect to $P \subseteq C$ if one of the following conditions holds:

- according to decision soft set $D = (F, d)$, object x belongs to Cl_t^{\geq} ,
- according to decision soft set $D = (F, d)$, object x does not belong to Cl_t^{\geq} , but it is inconsistent in the sense of the dominance principle with an object y belonging to Cl_t^{\geq} .

In terms of ambiguity, x possibly belongs to Cl_t^{\geq} with respect to $P \subseteq C$, if x possibly belongs to Cl_t^{\geq} with or without ambiguity. Due to the reflexivity of the soft dominance relation D_p , the above conditions can be summarized as follows: x possibly belongs to class Cl_t or better, with respect to $P \subseteq C$, if among the objects P_{soft} -dominated by x there is an object y belonging to class Cl_t or better, i.e., $D_p^-(x) \cap Cl_t^{\geq} \neq \emptyset$.

The P_{soft} -lower approximation of Cl_t^{\geq} , denoted by $\underline{P}(Cl_t^{\geq})$, and the P_{soft} -upper approximation of Cl_t^{\geq} , denoted by $\overline{P}(Cl_t^{\geq})$, are defined as follows ($t=1, \dots, n$):

$$\underline{P}(Cl_t^{\geq}) = \{x \in U : D_p^+(x) \subseteq Cl_t^{\geq}\},$$

$$\overline{P}(Cl_t^{\geq}) = \{x \in U : D_p^-(x) \cap Cl_t^{\geq} \neq \emptyset\}.$$

Analogously, one can define the P_{soft} -lower approximation and the P_{soft} -upper approximation of Cl_t^{\leq} as follows ($t=1, \dots, n$):

$$\underline{P}(Cl_t^{\leq}) = \{x \in U : D_p^-(x) \subseteq Cl_t^{\leq}\},$$

$$\overline{P}(Cl_t^{\leq}) = \{x \in U : D_p^+(x) \cap Cl_t^{\leq} \neq \emptyset\}.$$

The P_{soft} -lower and P_{soft} -upper approximation defined above, satisfy the following properties for each $t \in \{1, \dots, n\}$ and for any $P \subseteq C$:

$$\underline{P}(Cl_t^{\geq}) \subseteq Cl_t^{\geq} \subseteq \overline{P}(Cl_t^{\geq}), \underline{P}(Cl_t^{\leq}) \subseteq Cl_t^{\leq} \subseteq \overline{P}(Cl_t^{\leq}).$$

The P_{soft} -lower and P_{soft} -upper approximations of Cl_t^{\geq} and Cl_t^{\leq} have an important complementary property, according to which,

$$\begin{aligned} \underline{P}(Cl_t^{\geq}) &= U - \overline{P}(Cl_{t-1}^{\leq}) \text{ and } \overline{P}(Cl_t^{\geq}) = U - \underline{P}(Cl_{t-1}^{\leq}), t = 2, \dots, n, \\ \underline{P}(Cl_t^{\leq}) &= U - \overline{P}(Cl_{t+1}^{\geq}) \text{ and } \overline{P}(Cl_t^{\leq}) = U - \underline{P}(Cl_{t+1}^{\geq}), t = 1, \dots, n-1. \end{aligned}$$

The P_{soft} -boundaries of Cl_t^{\geq} and Cl_t^{\leq} , denoted by $Bn_p(Cl_t^{\geq})$ and $Bn_p(Cl_t^{\leq})$ respectively, and defined as follows ($t=1, \dots, n$):

$$Bn_p(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq}), \text{ and } Bn_p(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq}).$$

Due to complementary property, $Bn_p(Cl_t^{\geq}) = Bn_p(Cl_{t-1}^{\leq})$, for $t=2, \dots, n$.

For any criterion soft set $P \subseteq C$, we define the accuracy of approximation of Cl_t^{\geq} and Cl_t^{\leq} for all $t \in T$ respectively as

$$\alpha_p(Cl_t^{\geq}) = \frac{|P(Cl_t^{\geq})|}{|\overline{P}(Cl_t^{\geq})|}, \quad \alpha_p(Cl_t^{\leq}) = \frac{|\underline{P}(Cl_t^{\leq})|}{|\overline{P}(Cl_t^{\leq})|}.$$

The quality of approximation of the ordinal classification Cl by a set of soft set P is defined as the ration of the number of objects P_{soft} -consistent with the dominance principle and the number of all objects in U . Since the P_{soft} -consistent objects are those which do not belong to any P_{soft} -boundary $Bn_p(Cl_t^{\geq})$, $t=2, \dots, n$, or $Bn_p(Cl_t^{\leq})$, $t=1, \dots, n-1$, the quality of approximation of the ordinal classification Cl by a set of soft set P , can be written as

$$\gamma_p(Cl) = \frac{\left| U - \left(\left(\bigcup_{t \in T} Bn_p(Cl_t^{\geq}) \right) \cup \left(\bigcup_{t \in T} Bn_p(Cl_t^{\leq}) \right) \right) \right|}{|U|}.$$

$\gamma_p(Cl)$ can be seen as a degree of consistency of the objects from U , where P is the set of criterion soft set and Cl is the considered ordinal classification. Every minimal subset $P \subseteq C$ such that $\gamma_p(Cl) = \gamma_c(Cl)$ is called a *reduct* of Cl and is denoted by RED_{Cl} . Moreover, for a given set of U one may have more than one *reduct*. The intersection of all *reducts* is known as the core, denoted by $CORE_{Cl}$.

The dominance-based soft approximations of upward and downward unions of classes can serve to induce “if . . . , then . . .” decision rules. It is therefore more meaningful to consider the following three types of decision:

1. D_{\succeq} - decision rules that have the following form:
if $f(x, q1) \geq r_{q1}$ and $f(x, q2) \geq r_{q2}$ and ... $f(x, qp) \geq r_{qp}$ then $x \in Cl_t^{\succeq}$, where $P = \{q1, q2, \dots, qp\} \subseteq C$, $(r_{q1}, r_{q2}, \dots, r_{qp}) \in V_{q1} \times V_{q2} \times \dots \times V_{qp}$ and $t \in T$. These rules are supported only by objects from the P_{soft} -lower approximation of the upward unions of classes Cl_t^{\succeq} .
2. D_{\preceq} - decision rules that have the following form:
if $f(x, q1) \leq r_{q1}$ and $f(x, q2) \leq r_{q2}$ and ... $f(x, qp) \leq r_{qp}$ then $x \in Cl_t^{\preceq}$, where $P = \{q1, q2, \dots, qp\} \subseteq C$, $(r_{q1}, r_{q2}, \dots, r_{qp}) \in V_{q1} \times V_{q2} \times \dots \times V_{qp}$ and $t \in T$. These rules are supported only by objects from the P_{soft} -lower approximation of the downward unions of classes Cl_t^{\preceq} .
3. $D_{\succeq\preceq}$ - decision rules that have the following form:
if $f(x, q1) \geq r_{q1}$ and $f(x, q2) \geq r_{q2}$ and ... $f(x, qk) \geq r_{qk}$ and $f(x, qk + 1) \leq r_{qk+1}$ and ... $f(x, qp) \leq r_{qp}$ then $x \in Cl_t \cup Cl_{t+1} \cup \dots \cup Cl_s$, where $P = \{q1, q2, \dots, qp\} \subseteq C$, $(r_{q1}, r_{q2}, \dots, r_{qp}) \in V_{q1} \times V_{q2} \times \dots \times V_{qp}$ and $s, t \in T$ such that $t < s$. These rules are supported only by objects from the P_{soft} -boundaries of the unions of classes Cl_t^{\preceq} and Cl_t^{\succeq} .

5 Experiments

On the basis of data proposed by Grabisch [5], for an evaluation in a high school, an example is taken to illustrate the application of the method. The director of the school wants to assign students to two classes: bad and good. To fix the classification rules the director is asked to present some examples. The examples concern with six students described by means of four attributes (see Table 2 below):

- A_1 : level in Mathematics,
- A_2 : level in Physics,
- A_3 : level in Literature,
- A_4 : global evaluation (decision class).

Table 2. Multi-valued information table with examples of classification

Student	A_1 (Mathematics)	A_2 (Physics)	A_3 (Literature)	A_4 (Evaluation)
1	good	good	bad	good
2	medium	bad	bad	bad
3	medium	bad	bad	good
4	bad	bad	bad	bad
5	medium	good	good	bad
6	good	bad	good	good

The components of the multi-valued information table S are:

$$\begin{aligned}
 U &= \{1,2,3,4,5,6\} \\
 A &= \{A_1, A_2, A_3, A_4\} \\
 V_1 &= \{bad, medium, good\} \\
 V_2 &= \{bad, good\} \\
 V_3 &= \{bad, good\} \\
 V_4 &= \{bad, good\}
 \end{aligned}$$

the information function $f(x, q)$, taking values $f(1, A_1)=good$, $f(1, A_2) = good$, and so on.

Within this approach we approximate the class Cl_1^{\leq} of “(at most) bad” students and the class Cl_1^{\geq} of “(at least) good” students. Since information table as in table 1 above consist of only two decision classes, we have $Cl_1^{\leq} = Cl_1$, and $Cl_2^{\geq} = Cl_2$. Moreover, $C = \{A_1, A_2, A_3\}$ and $D = \{A_4\}$. In this case, however, A_1, A_2 and A_3 are criteria and the classes are preference-ordered. Furthermore, the multi-soft set equivalent to the multi-valued information table as in table 1 is given below:

$$(F, A) = \begin{cases} (F, a_1) = \{bad = \{4\}, medium = \{2,3,5\}, good = \{1,6\}\} \\ (F, a_2) = \{bad = \{2,3,4,6\}, good = \{1,5\}\} \\ (F, a_3) = \{bad = \{1,2,3,4\}, good = \{5,6\}\} \\ (F, d) = \{bad = \{2,4,5\}, good = \{1,3,6\}\} \end{cases}$$

And, with respect to each criterion soft set:

- $(F, a_1) : p_1 = bad, p_2 = medium, \text{ and } p_3 = good;$
- $(F, a_2) : p_1 = bad, \text{ and } p_2 = good;$
- $(F, a_3) : p_1 = bad, \text{ and } p_2 = good;$ and
- $(F, d) : p_1 = bad, \text{ and } p_2 = good.$

Our experiment obtained the following results. The P_{soft} -lower approximations, P_{soft} -upper approximations and the P_{soft} -boundaries of classes Cl_1^{\leq} and Cl_2^{\geq} are equal to, respectively:

$$\begin{aligned}
 \underline{P}(Cl_1^{\leq}) &= \{4\}, \quad \overline{P}(Cl_1^{\leq}) = \{2,3,4,5\}, \quad Bn_p(Cl_1^{\leq}) = \{2,3,5\}, \\
 \underline{P}(Cl_2^{\geq}) &= \{1,6\}, \quad \overline{P}(Cl_2^{\geq}) = \{1,2,3,5,6\}, \quad Bn_p(Cl_2^{\geq}) = \{2,3,5\}.
 \end{aligned}$$

Therefore, the accuracy of the approximation is 0.25 for Cl_1^{\leq} and 0.4 for Cl_2^{\geq} , while the quality of approximation is equal to 0.5. There are only one reduct which is also the core, i.e. $Red_{Cl} = Core_{Cl} = \{A_1\}$.

The minimal set of decision rules that are derived from the experiment, are shown below (within the parenthesis are the objects that support the corresponding rule):

1. if $f(x, A_1) \geq \text{good}$, then $x \in Cl_2^{\geq}$ (1,6)
2. if $f(x, A_1) \leq \text{bad}$, then $x \in Cl_1^{\leq}$ (4)
3. if $f(x, A_1) \geq \text{medium}$ and $f(x, A_1) \leq \text{medium}$ (i.e. $f(x, A_1)$ is medium), then $x \in Cl_1 \cup Cl_2$ (2,3,5).

Let us notice from table 1 that student 5 dominates student 3, i.e. student 5 is at least as good as student 3 with respect to all the three criteria, however, student 5 has a global evaluation worse than student 3. Therefore, this can be seen as an inconsistency revealed by the approximation based on soft-dominance which cannot be captured by the approximation based on the mapping of parameters to the set of objects under consideration. Moreover, let us remark that the decision rules induced from approximation obtained from soft-dominance relations give a more synthetic representation of knowledge contained in the decision system.

6 Conclusion

In this paper we have presented the applicability of soft set theory in multi-criteria decision analysis (MCDA). Dominance-based soft set approach is an extension of soft set theory for MCDA which permits the dealing of typical inconsistencies during the consideration of criteria or preference-ordered decision classes. Based on the approximations obtained through the soft-dominance relation, it is possible to induce a generalized description of the preferential information contained in the decision system, in terms of decision rules. The decision rules are expressions of the form if (condition) then (consequent), represented in a form of dependency between condition and decision criteria. When the proposed approach is applied to the multi-valued decision system through simulation, the decisions rules obtained are equivalent to the one that obtained by the previous technique [7] using rough-set based approach.

References

1. Albadvi, A., Chaharsooghi, S., Esfahanipour, A.: Decision making in stock trading: An application of PROMETHEE. *European Journal of Operational Research* 177(2), 673–683 (2007)
2. Chaudhuri, A., De, K., Chatterjee, D.: Solution of decision making problems using fuzzy soft relations. *Int. J. Information Technology* 15(1), 78–107 (2009)
3. Feng, F., Jun, Y.B., Liu, X., Li, L.: An adjustable approach to fuzzy soft set based decision making. *Journal of Computational & Applied Mathematics* 234, 10–20 (2010a)
4. Feng, F., Li, Y., Violeta, L.-F.: Application of level soft sets in decision making based on interval-valued fuzzy soft sets. *Journal of Computers and Mathematics with Applications* 60, 1756–1767 (2010b)
5. Grabisch, M.: Fuzzy integral in multicriteria decision making. *Fuzzy Sets and System* 89, 279–298 (1994)

6. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation of a preference relation in a pairwise comparison table. In: *Rough Sets in Knowledge Discovery: Applications, Case Studies, and Software Systems*, p. 13 (1998)
7. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research* 129(1), 1–47 (2001)
8. Herawan, T., Deris, M.M.: On Multi-soft Sets Construction in Information Systems. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) *ICIC 2009. LNCS (LNAI)*, vol. 5755, pp. 101–110. Springer, Heidelberg (2009)
9. Hu, Q., Xie, Z., Yu, D.: Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. *Pattern Recognition* 40(12), 3509–3521 (2007)
10. Isklar, G., Buyukozkan, G.: Using a multi-criteria decision making approach to evaluate mobile phone alternatives. *Computer Standard & Interfaces* 29(2), 265–274 (2007)
11. Jiang, Y., Tang, Y., Chen, Q.: An adjustable approach to intuitionistic fuzzy soft sets based decision making. *Journal of Applied Mathematical Modelling* 35, 824–836 (2011)
12. Maji, P.K., Roy, A.R., Biswas, R.: An application of soft sets in a decision making problem. *Computer and Mathematics with Application* 44, 1077–1083 (2002)
13. Molodtsov, D.: Soft set theory-first results. *Computers and Mathematics with Applications* 37, 19–31 (1999)
14. Pawlak, Z.: *Rough sets: Theoretical aspects of reasoning about data*. Springer, Heidelberg (1991)
15. Roy, A.R., Maji, P.K.: A fuzzy soft set theoretic approach to decision making problem. *Journal of Computational & Applied Mathematics* 203, 412–418 (2007)
16. Slowiński, R.: Rough Set Approach to Knowledge Discovery about Preferences. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS (LNAI)*, vol. 5796, pp. 1–21. Springer, Heidelberg (2009)
17. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters* 28(4), 459–471 (2007)
18. Yu, L., Wang, S., Lai, K.: An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring. *European Journal of Operational Research* 195(3), 942–959 (2009)
19. Zdzislaw, P.: Rough sets. *International Journal of Computer and Information Sciences* 11(5), 341–356 (1982)

Efficient Computation of Measurements of Correlated Patterns in Uncertain Data*

Lisi Chen¹, Shengfei Shi^{2,**}, and Jing Lv¹

¹ Honors School

² School of Computer Science and Technology,
Harbin Institute of Technology,
No.92 West Dazhi Street, Harbin

{chen.lisi,lv.jing}@yahoo.com.cn, shengfei@hit.edu.cn

Abstract. One of the most important tasks in data mining is to discover associations and correlations among items in a huge database. In recent years, some studies have been conducted to find a more accurate measure to describe correlations between items. It has been proved that the newly developed measures of all-confidence and bond perform much better in reflecting the true correlation relationship than just using support and confidence in categorical database. Hence, several efficient algorithms have been proposed to mine correlated patterns based on all-confidence and bond. However, as the data uncertainty become increasingly prevalent in various kinds of real-world applications, we need a brand new method to mine the true correlations in uncertain datasets with high efficiency and accuracy. In this paper, we propose effective methods based on dynamic programming to compute the expected all-confidence and expected bond, which could serve as a slant in finding correlated patterns in uncertain datasets.

Keywords: correlated patterns all-confidence bond uncertain data.

1 Introduction

Over the past few years, mining of associative rules has been widely studied in the field of data mining. Some popular algorithms with great efficiency and simplicity were proposed sequentially, such as Apriori and FP-Growth. Most of the algorithms use support and confidence as criteria in measuring interests of mined rules. However, the measurement of support and confidence are just inclined to reflect frequency of itemset. It is likely that some highly correlated association involving items with support value lower than minimum threshold would be considered uninteresting. In other words, with a high minimum support threshold, we could only find commonsense knowledge. But if we set a relatively low value of minimum support threshold, numerous redundant and

* This work is supported in part by the National Natural Science Foundation of China under grant 60703012; the Key Program of National Natural Science Foundation of China under grant 60933001.

** Corresponding author.

less informative association rules would be generated. In order to solve the problem, measurements of all-confidence and bond have been proposed as alternative interest measures for mining the true correlation relationships among a huge database. With the downward closure property, both of them can be computed efficiently.

Recently, several effective algorithms have been developed for mining correlated patterns based on measurements of all-confidence and bond. CoMine[4] and CcMine[5] are the most typical ones, which use all-confidence to evaluate the correlation of patterns. Both of them are based on the extensions of the FP-Growth methodology. Experimental results have shown that both CoMine and CcMine algorithm perform well in comparison with the counterpart Apriori-based algorithms.

In these years, uncertain data mining have been arousing much more attentions in relevant research areas due to the growing need to handle data uncertainty caused by noise, imprecise measurement, decision errors, etc in various kinds of real-world applications. Some algorithms have already been devised to mine the frequent itemset among uncertain dataset, such as the work of [6] and [7], where each item appears with particular probability. Some other algorithms like [10] and [11] were proposed for classifying uncertain data. However, it is still an open problem to develop algorithms for mining truly correlated patterns with data uncertainty.

In this paper, we propose an efficient method to compute the measurements of correlated patterns. Based on the possible world semantics, we make definitions for expected all-confidence and expected bond. According to the definitions, our new developed algorithms have been proved to be accurate and efficient. Our contributions could be generalized as follows.

- We are the first to propose the definition of expected all-confidence and expected bond as the measurement of correlation degree in uncertain database, which is proved to be accurate and effective in reflecting the correlation degree among items with data uncertainty.
- We develop novel DP-based methods to calculate expected all-confidence and expected bond with high efficiency.

For the rest part of this paper, the related work is introduced in Section 2. Preliminaries are stated in Section 3. The definitions of expected all-confidence and expected bond are stated in Section 4. Efficient computation of measurements of correlated patterns is devised in Section 4. The evaluation and experiment part is presented in Section 5. The conclusion is in Section 6.

2 Related Work

In the field of uncertain data mining, some algorithms that mine frequent patterns in uncertain database have been proposed in recent years. U-Apriori was proposed in [6], which uses expected support to discover all frequent itemsets in uncertain database. After that, UF-Growth was proposed in [7], of which data structure is based on the expansion of FP-Tree. Compared with U-Apriori, UF-Growth could avoid generating a large number of candidate itemsets. [8] introduced the frequent patterns mining in

uncertain data and how to expand more algorithms into uncertain field. To avoid the inaccuracy of using expected support to measure the frequency of itemsets, [2] proposed probabilistic frequent itemset mining algorithm with great efficiency using possible world semantics. Based on the HARMONY classification algorithm proposed in [9], [1] devised a classification algorithm in uncertain data named uHARMONY. In addition, [1] also devise an effective way to compute expected confidence based on dynamic programming.

3 Preliminaries

3.1 The Model for Uncertain Data

In this paper, we adopt the same model of uncertain data used in [2], which is based on the possible worlds semantic with existence of uncertain items.

Definition 1. Let I be a set of all possible items in transactional database T . If item $x \in I$ appears in transaction $t \in T$ with probability $P(x \in t) \in (0,1)$, then item x is an uncertain item.

It is obvious that certain item is a special case of uncertain item since $P(x \in t) \in \{0,1\}$. The definition of uncertain transactional databases is stated as follow.

Definition 2. Transaction t is an uncertain transaction if and only if t contains uncertain items. Transaction database T is an uncertain transaction database if and only if T contains uncertain transactions.

Then, we will present the possible worlds semantics. An uncertain transactional database could generate numerous possible worlds. Each possible world consists of a set of certain transactions. In other words, possible world is one possible combination of appearance of all items in transactional database. We could find that each uncertain item $x, 0 < P(x \in t_i) \leq 1$ would derive two possible worlds. One is x exists in t_i , the other is x does not exists in t_i . As a result, the number of possible worlds in uncertain transactional database would increase exponentially as the growing number of uncertain items.

We suppose that each two items are mutually independent. Then, we could calculate the appearance probability of each possible world through formula 1.

$$P(w) = \prod_{t \in I} \left(\prod_{x \in t} P(x \in t) \times \prod_{x \notin t} (1 - P(x \in t)) \right) \quad (1)$$

3.2 All-Confidence and Bond

According to the work of [3] and [4], all-confidence and bond are defined as follows.

Definition 3. Let X be an itemset and i_j be a subset of X containing only one item. The all-confidence of X is:

$$all_confidence(X) = \frac{sup(X)}{MAX\{sup(i_j) | i_j \in X\}}$$

The denominator of $all_confidence(X)$ is the maximum count of transactions that contain any single item of X . It could be easily deduced that all-confidence is the smallest confidence of any rule among the itemset X . Consequently, all of the rules generated from X would have confidence values no less than its all-confidence.

Definition 4. Let i_j be a subset of X containing only one item. T is a transaction database. The bond of X is defined as:

$$bond(X) = \frac{sup(X)}{|\{t_i \in T | \exists i_j \in X (i_j \in t_i)\}|}$$

We could consider bond to be the ratio of the support value of X , which is the number of transactions that contain all items in T , and the cardinality of the union of transactions that contain any item of X .

It has been proved in [3] that the relationship between all-confidence and bond satisfies the following formula

$$all_conf(X) \geq bond(X)$$

In addition, both all-confidence and bond hold the downward closure property. That is, if $X' \subset X$, then

$$\begin{aligned} all_conf(X') &\geq all_conf(X) \\ ond(X') &\geq bond(X) \end{aligned}$$

4 Efficient Computation of Expected All-Confidence and Bond

4.1 Definition of Expected All-Confidence and Bond

Since the definition of expected all-confidence and expected bond have never been proposed before, it is important to provide them with a proper definition. The model of possible worlds is widely accepted in the field of uncertain data mining. Consequently, we define the measurements of correlated patterns based on the possible worlds semantics.

At first, we define that $P(w_j)$ stands for the appearance probability of possible world w_j , $sup_{w_j}(X)$ stands for the support of X under possible world w_j , $universe_{w_j}(X)$ is the set of transactions satisfying the condition that transaction t includes either items in itemset X .

Before introducing the definition of expected all-confidence and bond, we will present the definition of expected support.

According to [2], expected support of itemset x is defined as the sum of the expected probabilities of presence of x in each of the transactions in the database. Hence, we have:

$$E(sup(x)) = \sum_{i=1}^{|T|} P(x, t_i) \tag{2}$$

Where $P(x, t_i)$ represent the probability of the itemset x occurring in a given transaction t_i .

Definition 5. The expected all-confidence of itemset X is:

$$E(all_conf(X)) = \sum_{w_j \in W} \frac{sup_{w_j}(X) \times P(w_j)}{MAX\{sup_{w_j}(x_i) \mid x_i \in X\}}$$

Now we present the definition of universe according to [4] before introducing the definition of expected bond.

Definition 6. Let $X = \{i_1, i_2, i_3, \dots, i_k\}$ be a set of items. T is a transaction database. The universe of X is defined as:

$$universe(X) = \{t_i \in T \mid \exists i_j \in X (i_j \in t_i)\}$$

Definition 7. The expected bond of itemset X is:

$$E(bond(X)) = \sum_{w_j \in W} \frac{sup_{w_j}(X) \times P(w_j)}{|universe_{w_j}(X)|}$$

From the definitions above, we could find that the expected all-confidence or the expected bond of X cannot be converted into the following ratio:

$$E(sup(X))/MAX\{E(sup(x_i)) \mid x_i \in X\}, \quad E(sup(X))/E(|universe(X)|)$$

It is obvious that the values of the expressions above are not equal to the corresponding definitions.

Actually, it is impossible to compute the expected all-confidence or expected bond using Exhaustive Attack method based on the model of possible worlds due to the huge number of possible worlds. Hence, it is quite necessary to devise new efficient algorithms to solve the problem.

4.2 Efficient Computation of Expected Bond

Firstly, we convert the definition of expected bond into a form that is easier to compute using dynamic programming.

Lemma 1. Since $0 \leq sup(X) \leq |universe(X)| \leq |T|$, we have:

$$\begin{aligned}
 E(\text{bond}(X)) &= \sum_{w_j \in W} \frac{\text{sup}_{w_j}(X) \times P(w_j)}{|\text{universe}_{w_j}(X)|} \\
 &= \sum_{k=0}^{|T|} \sum_{i=0}^k \frac{i}{k} \times P(\text{sup}(X) = i \wedge |\text{universe}(X)| = k) \\
 &= \sum_{k=0}^{|T|} \frac{E_{\text{sup}(x)=k}(\text{sup}(X))}{k}
 \end{aligned}$$

Because the computation of expected bond bears the properties of dynamic programming, namely, overlapping subproblems and optimal substructure, we can use the method to compute expected bond.

We define that $E_{k,j}(\text{bond}(X))$ is the expected bond on first k transactions of T when $|\text{universe}(X)|=k$. $t_k \in T$ represents the k -th transaction of T . Then, we have the following theorem.

Theorem 1. Given $0 \leq k \leq |T|$, we have:

$$\begin{aligned}
 E_{k,j}(\text{bond}(X)) &= P(X \subseteq t_k) \times E_{k-1,j-1}(\text{bond}(X) + \frac{1}{|\text{universe}(X)|}) \\
 &\quad + P((x \subseteq t_k \mid x \in \text{universe}(X)) \wedge X \not\subseteq t_k) \times E_{k-1,j-1}(\text{bond}(X)) \\
 &\quad + P(x \not\subseteq t_k \mid x \in \text{universe}(X)) \times E_{k,j-1}(\text{bond}(X))
 \end{aligned}$$

Proof. If $X \subseteq t_k$, we have:

$$E_{k,j}(\text{bond}(X)) = P(X \subseteq t_k) \times E_{k-1,j-1}(\text{bond}(X) + \frac{1}{|\text{universe}(X)|})$$

since $E_{k,j}(\text{bond}(X))$ is $1/|\text{universe}(X)|$ more than $E_{k-1,j-1}(\text{bond}(X))$ in each possible world.

When $x \in \text{universe}(X)$, if $x \subseteq t_k$ but $X \not\subseteq t_k$, we have:

$$E_{k,j}(\text{bond}(X)) = E_{k-1,j-1}(\text{bond}(X))$$

since $E_{k,j}(\text{bond}(X))$ remains the same as $E_{k-1,j-1}(\text{bond}(X))$ in each possible world.

When $x \in \text{universe}(X)$, if $x \not\subseteq t_k$, then we have:

$$E_{k,j}(\text{bond}(X)) = E_{k,j-1}(\text{bond}(X))$$

since all the subset of X do not belong to t_k . Hence, $E_{k,j}(\text{bond}(X))$ remains the same as $E_{k,j-1}(\text{bond}(X))$ in each possible world.

Thus, Theorem 1 is proved.

According to the recursion formula, we could find that

$$E_{k-1,j-1}(bond(X) + 1) = E_{k-1,j-1}(bond(X)) + P_{k-1,j-1}(X)$$

$P_{k-1,j-1}(X)$ is defined as the probability of satisfying $universe(X)|=k$ on first j transactions in T .

In order to compute $P_{k,j}(X)$, we adopt the dynamic programming methodology similar to [1], here is the recursive formula:

$$P_{k,j}(X) = P(X \subseteq t_k) \times P_{k-1,j-1}(X) + (1 - P(X \subseteq t_k)) \times P_{k,j-1}(X)$$

The initial conditions are $P_{0,0}(X) = 1$ $P_{0,j}(X) = (1 - P(X \subseteq t_k)) \times P_{0,j-1}(X)$

We ellipse the proof since it is similar to that of Theorem 1.

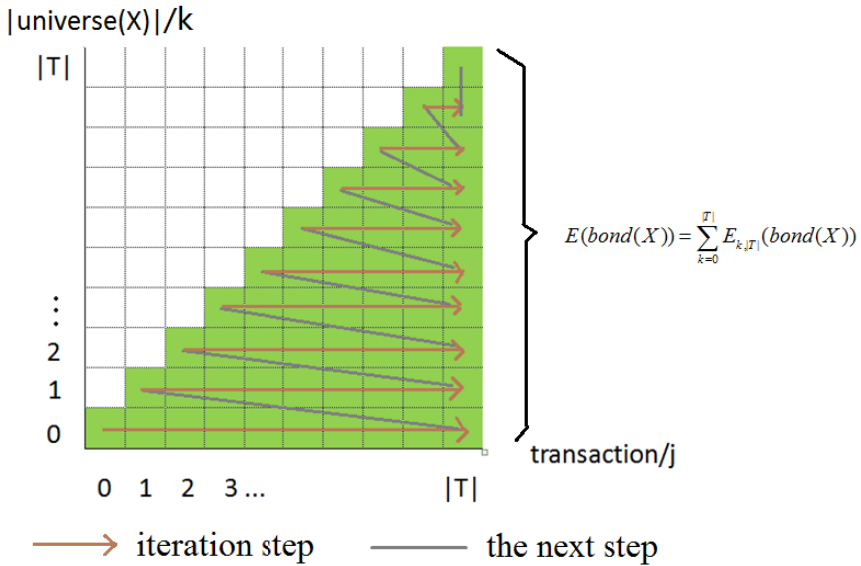


Fig. 1. Computation process of expected bond

From Figure 1, it is easy to conclude that the computation of the expected bond requires at most $O(|T|^2)$ time and at most $O(|T|)$ space.

4.3 Efficient Computation of All-Confidence

Since the computation of all-confidence involves calculating the MAX value of support in denominator, we cannot use dynamic programming, which requires the properties of overlapping subproblems and optimal substructure, to simplify the time complexity. However, we can still acquire a partial recurrence formula through converting the expression of all-confidence.

While computing expected all-confidence under possible world model, the item with the maximum support value in an itemset could be different. Therefore, it cannot

fulfill the prerequisites of dynamic programming. To eliminate such negative effect, we divide the set of all possible worlds into some conditional possible worlds sets. Each item in an itemset appears with a particular frequency. For example , if $X=\{x_1, x_2, x_3\}$ and the frequencies of x_1, x_2 and x_3 are 5, 3, 6, respectively. Then, this condition could be a conditional possible worlds set.

We use W_C to represent conditional possible worlds set.

Definition 8. Let itemsets $A_1, A_2, \dots, A_{|X|}$ be $\{1, 2, \dots, |T|\}$, $|T|$ is the number of transactions in uncertain transactional database T , $|X|$ is the number of items in itemset $X = \{x_1, x_2, \dots, x_n\}$. \mathbb{A} is the Cartesian product of $A_1, A_2, \dots, A_{|X|}$, namely $\mathbb{A} = A_1 \times A_2 \times \dots \times A_{|X|}$. $\alpha = (a_1, a_2, \dots, a_n)$ is an element in \mathbb{A} , of which $a_1 \in A_1, a_2 \in A_2, \dots, a_n \in A_n$. Then, $W_C(\alpha)$ stands for the conditional possible worlds set satisfying the following condition: the frequency of x_1 in T is a_1 , the frequency of x_2 in T is a_2 , ..., the frequency of x_n in T is a_n .

Similar to the computation of expected bond, the expression of all-confidence in definition 8 could be converted as follow, of which $E_{W_C(\alpha)}(all_conf(X))$ stands for the expected all-confidence of X under the conditional possible worlds set $W_C(\alpha)$:

$$E(all_conf(X)) = \sum_{\alpha \in \mathbb{A}} E_{W_C(\alpha)}(all_conf(X)) \times P(W_C(\alpha)) \tag{3}$$

Since the item of X which has the maximum support remains the same under the same conditional possible worlds set, properties of overlapping subproblems and optimal substructure could be satisfied. As a result, we are able to apply dynamic programming methodology to compute expected all-confidence based on formula 3.

According to definition 8, let $n=|X|$ and $B_1 \subseteq A_1, B_2 \subseteq A_2, \dots, B_n \subseteq A_n$, of which $B_1 = \{b_1, b_1 - 1\}, B_2 = \{b_2, b_2 - 1\}, \dots, B_n = \{b_n, b_n - 1\}$, $\mathbb{B} = B_1 \times B_2 \times \dots \times B_n$. To simplify the computation, we introduce a function f . The independent variable of f is the conditional possible worlds set α . The induced variable of f is a Boolean expression. Let $b_m = u_m, b_m - 1 = d_m$. The detailed mapping relations are stated as follows.

$$f(\alpha) = f(c_1, c_2, \dots, c_n) \quad c_1 = \{u_1, d_1\}, c_2 = \{u_2, d_2\}, \dots, c_n = \{u_n, d_n\}$$

If $c_m = u_m$, then generate Boolean expression $x_m \notin t_k$

If $c_m = d_m$, then generate Boolean expression $x_m \in t_k$

of which x_m is the m -th item of itemset X .

For example, if $|X|=4, \alpha = (u_1, d_2, d_3, u_4)$, then we have

$$f(\alpha) = f(u_1, d_2, d_3, u_4) = x_1 \notin t_k \wedge x_2 \in t_k \wedge x_3 \in t_k \wedge x_4 \notin t_k$$

Theorem 2. Let $\alpha \in \mathbb{B}$, $\bar{\alpha} \in \mathbb{B}$ and $\alpha = (b_1, b_2, \dots, b_n)$, $\bar{\alpha} = (b_1 - 1, b_2 - 1, \dots, b_n - 1)$ $E_{k, W_C(\alpha)}(all_conf(X))$ represents the expected all-confidence under conditional possible worlds sets $W_C(\alpha)$ in the first k transactions of T . Then we have:

$$\begin{aligned} & E_{k, W_C(\alpha)}(all_conf(X)) \\ &= (E_{k-1, W_C(\bar{\alpha})}(all_conf(X)) + \frac{1}{MAX\{q \mid q \in element(\alpha)\}}) \times P(X \subseteq t_k) \\ &+ \sum_{\beta \in (\mathbb{B} - \alpha)} E_{k-1, W_C(\beta)}(all_conf(X)) \times P(f(\beta)) \end{aligned}$$

Proof. When $X \subseteq t_k$, we have

$$E_{k, W_C(\alpha)}(all_conf(X)) = E_{k-1, W_C(\bar{\alpha})}(all_conf(X)) + \frac{1}{MAX\{q \mid q \in element(\alpha)\}}$$

since the expected all-confidence of first k transactions in T is $1/MAX\{q \mid q \in element(\alpha)\}$ greater than that of first $k-1$ transactions in T under conditional possible world $W_C(\alpha)$.

When $X \not\subseteq t_k$, for each items x_1, x_2, \dots, x_n in itemset X , some of them would belong to t_k while others would not. Hence, we need to consider all possible situations. If $x_m \in t_k$ and the frequency of x_m under conditional possible worlds set $W_C(\alpha)$ in first k transactions is b_m , the expected all-confidence under conditional possible worlds set $W_C(\alpha)$ in first $k-1$ transactions with the frequency of x_m being $b_m - 1$ should be inherited. Else if $x_m \notin t_k$, the expected all-confidence under conditional possible worlds set $W_C(\alpha)$ in first $k-1$ transactions with the frequency of x_m being b_m should be inherited. Then, other items in itemset X are analyzed likewise. Hence, we have:

$$E_{k, W_C(\alpha)}(all_conf(X)) = \sum_{\beta \in (\mathbb{B} - \alpha)} E_{k-1, W_C(\beta)}(all_conf(X)) \times P(f(\beta))$$

Before using the recurrence formula, we should consider the initialization on condition that $k=1$. Such problem is very simple since it could be calculated directly through the definition 5.

Then, we unfold the recurrence formula just like that of expected bond. We have:

$$\begin{aligned} & E_{k, W_C(\alpha)}(all_conf(X)) \\ &= E_{k-1, W_C(\bar{\alpha})}(all_conf(X)) \times P(X \subseteq t_k) + \frac{P_{k-1, W_C(\bar{\alpha})}(X)}{MAX\{q \mid q \in element(\alpha)\}} \times P(X \subseteq t_k) \\ &+ \sum_{\beta \in (\mathbb{B} - \alpha)} E_{k-1, W_C(\beta)}(all_conf(X)) \times P(f(\beta)) \end{aligned}$$

Of which $P_{k-1, W_C(\bar{\alpha})}(X)$ is the probability of conditional possible worlds set $W_C(\bar{\alpha})$. $P_{k, W_C(\alpha)}(X)$ could be computed by dynamic programming, too. According to our previous suppose, we have:

$$P_{k,W_C(\alpha)}(X) = \sum_{\beta \in \mathbb{B}} P_{k-1,W_C(\beta)}(X) \times P(f(\beta)) \tag{4}$$

To get the final result, we should use the formula 3 to sum up the expected all-confidence multiplying by corresponding probability under each conditional possible worlds set.

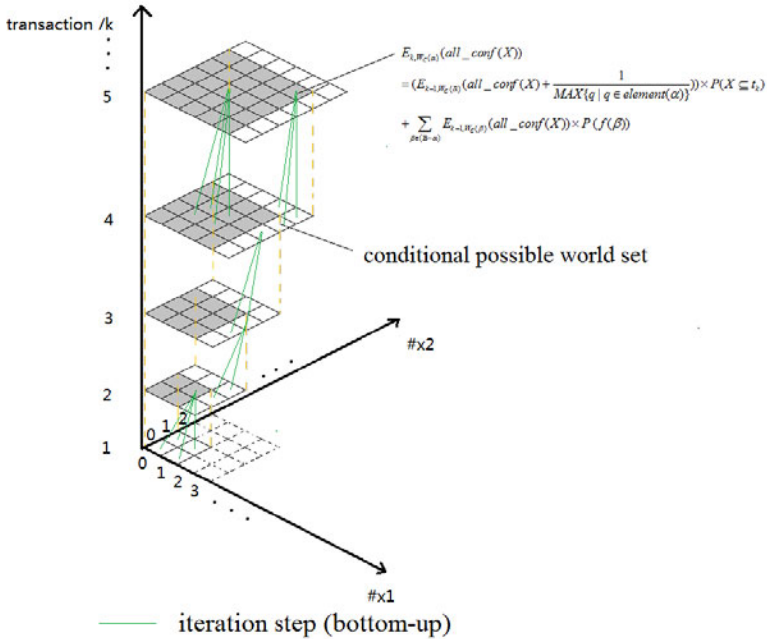


Fig. 2. Computation process of expected all-confidence

We offer an example in Figure 2, which is on condition that $|X|=2$, to illustrate the process of the computation algorithm.

In Figure 2, the coordinate #X1, #X2 represent the frequency of items x_1 and x_2 in itemset X , respectively. The ordinate represent the first k transactions in database. Hence, a small square stands for a conditional possible worlds set in first k transactions. $E_{k,W_C(\alpha)}(all_conf(X))$ is acquired through iteration, which requires $E_{k-1,W_C(\beta_2)}(all_conf(X))$, $E_{k-1,W_C(\beta_3)}(all_conf(X))$ and $E_{k-1,W_C(\beta_4)}(all_conf(X))$.

Of which:

$$\begin{cases} \alpha = (b_1, b_2) & b_1, b_2 \in [0, k] \\ \beta_1 = (b_1, b_2) & b_1, b_2 \in [0, k-1] \\ \beta_2 = (b_1-1, b_2) & b_1, b_2 \in [0, k-1] \\ \beta_3 = (b_1, b_2-1) & b_1, b_2 \in [0, k-1] \\ \beta_4 = (b_1-1, b_2-1) & b_1, b_2 \in [0, k-1] \end{cases}$$

Finally, we make an analysis about the time complexity and space complexity of the computation of expected all-confidence.

Theorem 3. The computation of the expected all-confidence requires at most $O(|T|^{|X|+1})$ time and at most $O(|T|^{|X|})$ space.

Proof. The number of iterations is dependent on the number of conditional possible worlds sets and the number of transactions. During the computation, we need to compute the expected all-confidence under each conditional possible world sets. For each distinct value k , the number of conditional possible worlds sets is $k^{|X|}$. Therefore, the number of all conditional possible worlds sets in the process of iteration is $1^{|X|} + 2^{|X|} + 3^{|X|} + \dots + |T|^{|X|}$. Since each iteration requires $O(1)$ time, the total time required is $O(|T|^{|X|+1})$.

As for the space complexity, we need to store the expected all-confidence value under each conditional possible world sets when $k=|T|$. As a consequence, the total space required is $O(|T|^{|X|})$.

5 Experiments and Evaluations

In this section, we will present the evaluation results of our algorithm for computing expected bond and expected all-confidence. Our algorithm is implemented in C++. All the experiments were conducted on a computer with Intel Core 2 Duo T9550 CPU (2.66GHz) and 4GB memory.

5.1 Evaluation Datasets

Since uncertain transactional datasets cannot be acquired directly at present, we get the data uncertainty by adding an appearance probability into the acquired certain transactional datasets. In this experiment, we first generate data without uncertainty using the IBM Quest Market-Basket Synthetic Data Generator. The whole datasets have 10000 transactions. Each transaction has 20 items. The number of all possible items in transactional database is 50000. At the same time, we would select first k transactions for partial datasets according to the requirement of evaluation. To avoid a low expected bond or expected all-confidence value, especially when the number of items goes large, we assign each item a relatively higher appearance probability. We suppose that the uncertainty of each item is subject to the gauss distribution $N(\mu, \sigma^2)$. For each item, $\mu \in [0.60, 0.99]$, $\sigma \in [0.05, 0.1]$. They both generated randomly. According to the given gauss distribution, we generate an appearance probability ranging from 0 to 1 for each item.

5.2 Evaluation Result

Firstly, we make an experiment on the computation of expected bond. During the experiment, we select first 1000, 2000, 3000, ..., 10000 transactions, respectively.

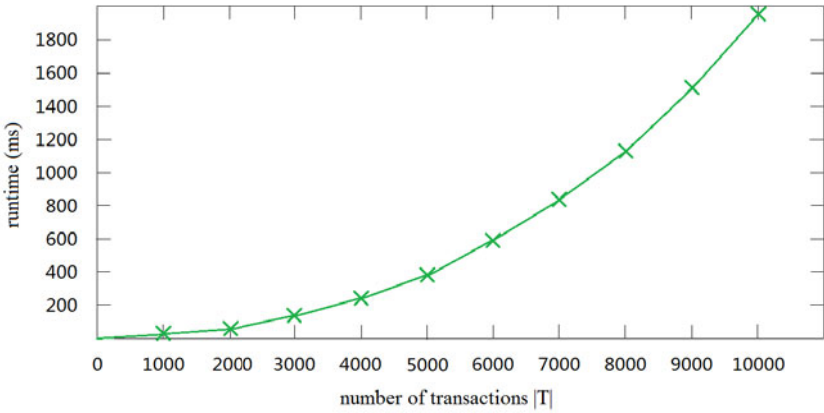


Fig. 3. Running time evaluation in computation of expected bond

Each condition includes 5 independent experiments with 2-itemset, 3-itemset, 4-itemset, 5-itemset and 6-itemset. Each itemset is generated randomly.

Figure 3 shows the relationship between the number of transactions and runtime. We could find that the time complexity is $O(|T|^2)$, which is in accordance with our previous conclusion.

Then, we make an experiment on the computation of expected all-confidence. The itemset in the experiment is 2-itemset.

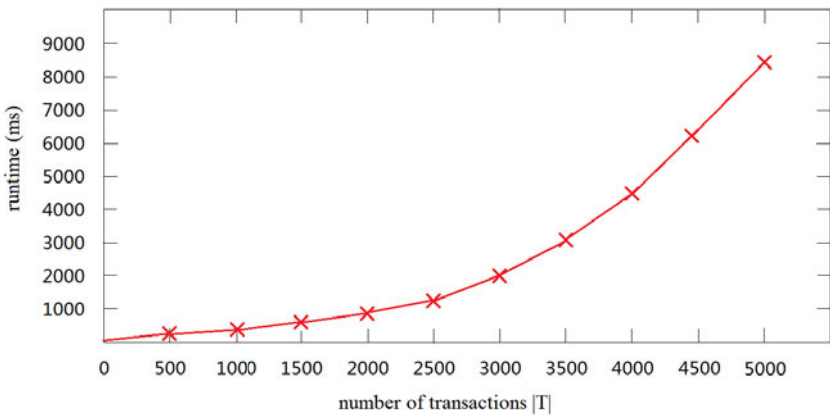


Fig. 4. Running time evaluation in the computation of all-confidence

Figure 4 shows the relationship between the number of transactions and runtime. We could find that the time complexity is $O(|T|^3)$, which is in accordance with Theorem 3.

Since the computation of expected all-confidence is dependent on the number of items in itemset, experiment reflecting the relationship between the number of items and runtime is also proposed. We let the number of transactions be 1000. Then we

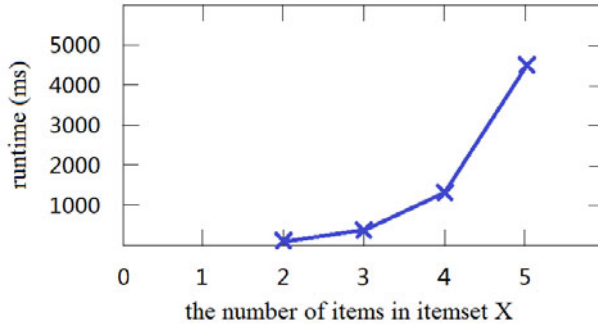


Fig. 5. Runtime evaluation in computation of all-confidence regarding to $|X|$

choose 25 itemsets with 5 different numbers of items, respectively. Each item in a particular itemset is generated randomly.

From Figure 5, we could find that the runtime increases exponentially as the number of items in itemset X increases.

However, the number of items in itemset, to a great extent, is limited since the boost of $|X|$ would lead the all-confidence value to reduce dramatically. As a result, the algorithm should consider to be effective as long as $|X| \ll |I|$.

6 Conclusions

In this paper we propose brand new algorithms based on dynamic programming to compute expected bond and expected all-confidence in uncertain data. Since the two algorithms are devised under possible world semantics, we could acquire the exact value of expected bond and expected all-confidence. Compared with the Exhaustive Attack method based on the definition of possible world with the time of $O(2^{|X||I|})$, the new algorithms are able to reduce the computation of expected bond and expected all-confidence to $O(|I|^2)$ and $O(|I|^{|X|+1})$, respectively. Consequently, our algorithms have been proved to be accurate and efficient.

References

1. Gao, C., Wang, J.: Direct Mining of Discriminative Patterns for Classifying Uncertain Data. In: Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 861–870 (2010)
2. Bernecker, T., Riegel, H., Renz, M., Verhein, F., Zuefle, A.: Probabilistic Frequent Itemset Mining in Uncertain Databases. In: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 119–128 (2009)
3. Omiecinski, E.R.: Alternative Interest Measures for Mining Associations in Databases. IEEE Transactions on Knowledge and Data Engineering, TKDE 15, 57–69 (2003)

4. Lee, Y.K., Kim, W.Y., Cai, Y.D., Han, J.: CoMine: Efficient Mining of Correlated Patterns. In: Proceedings of 3rd IEEE International Conference on Data Mining, ICDM 2003, Melbourne, FL, USA, pp. 581–584 (2003)
5. Kim, W.-Y., Lee, Y.-K., Han, J.: CcMine: Efficient Mining of Confidence-Closed Correlated Patterns. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 569–579. Springer, Heidelberg (2004)
6. Chui, C.-K., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
7. Leung, C.K.S., Carmichael, C.L., Hao, B.: Efficient Mining of Frequent Patterns from Uncertain Data. In: Workshops Proceedings of the 7th IEEE International Conference on Data Mining, PAKDD 2007, Omaha, Nebraska, USA, pp. 489–494 (2007)
8. Aggarwal, C.C., Li, Y., Wang, J., Wang, J.: Frequent pattern mining with uncertain data. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 29–38 (2009)
9. Wang, J., Karypis, G.: On mining instance-centric classification rules. *IEEE Transactions on Knowledge and Data Engineering*, TKDE 18, 1497–1511 (2006)
10. Qin, B., Xia, Y., Li, F.: DTU: A Decision Tree for Uncertain Data. In: Theeramunkong, T., Kijssirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 4–15. Springer, Heidelberg (2009)
11. Qin, B., Xia, Y., Prabhakar, S., Tu, Y.-C.: A rule-based classification algorithm for uncertain data. In: Proceedings of the IEEE 25th International Conference on Data Engineering, ICDE 2009, Shanghai, China, pp. 1633–1640 (2009)

Efficient Subject-Oriented Evaluating and Mining Methods for Data with Schema Uncertainty^{*}

Yue Wang^{1,**}, Changjie Tang², Tengjiao Wang¹,
Dongqing Yang¹, and Jun Zhu³

¹ Key Laboratory of High Confidence Software Technologies (Peking University),
Ministry of Education, China

{eecswangyue,tjwang,dqyang}@pku.edu.cn

² School of Computer Science, Sichuan University, Chengdu, 610065, China

³ China Bith Defect Monitoring Centre, Sichuan University, Chengdu, 610065, China

Abstract. With the progressing of data collecting methods, people have already collected scales of data for various application fields such as medical science, meteorology, electronic commerce and so on. To analyze these data needs to integrate data from the various heterogeneous data sets. As historical reasons technically or non-technically, usually, the schemas of the data sets to be integrated are complex and different. Thus to analyze the integrated data may cause ambiguous results for their non-uniform schemas. This paper targets mining this kind of data, and its main contributions include:(1) proposed schema uncertainty to describe data with non-uniform schemas and proposed couple correlation degree (Cor) to evaluate the existence probabilities for records in data with schema uncertainty based on the analyzing subject;(2) designed a data structure "B-correlation tree" to establish a hierarchical structure for uncertain data with their existence probabilities and discussed the distribution affection by selecting nodes on different levels of B-correlation tree ; (3) proposed a efficient Monte Carlo uncertain data analyzing algorithm, MonteCarlo-evaluate (MCE), based on B-correlation tree for data with schema uncertainty; (4) analyzed the accuracy and convergence property for MCE theoretically; (5) implemented a prototype system by using B-correlation tree and MCE on real medical data and synthetic TPC-H benchmark [20] data; provided sufficient experiments to test the effectiveness and efficiency of the provided methods. The results of experiments show that: the provided methods can efficient evaluate the schema uncertainty in data and thus can be equal to the tasks of analyzing large scale data with schema uncertainty efficiently.

Keywords: schema uncertainty, uncertain data mining, Monte Carlo method.

^{*} This work is supported by the National Key Technology R&D Program of China(No. 2009BAK63B08), National High Technology Research and Development Program of China('863' Program)(No.2009AA01Z150), National Science& Technology Pillar Program of China(No. 2009BAH44B03), China Postdoctoral Science Foundation.

^{**} Corresponding author.

1 Introduction

The background of this research is based on the task of the birth defect medical data mining for China Birth Defect Monitoring Centre (CBDMC). To analyze the relations between the possible factors and various birth defect diseases, CBDMC has started a project to collect birth defect medical data since 1970s. And to fulfill this mission, CBDMC cost a lot of time and money to collect data from local hospitals. Thus these data are very valuable both for their time continuity and money consuming. Medical experts want to make use of these data to evaluate the effect of the former measures taken by government to reduce the birth defect rate. As the data collecting methods and standards changed a lot during the years from 1970s to 2010s, these historical data are not always maintain a uniform data schema. Simply analyze this kind of data by cleaning its non-uniform part may cause the results biased from the fact a lot (for over 20% of the original data have different schemas to the others, many valuable knowledge may be contained in the non-uniform parts of data), thus people need method to analyze this kind of data. Uncertain data analysis is a good way to analyze data with uncertainty, and there are several challenges to apply it.

Challenge 1. Evaluate the uncertainty for data records: we use correlation degree between every data record and the analysis subject to estimate the uncertainty of data. When the scale of data is large, to evaluate that correlation degree efficiently for every data records in data set challenges a lot.

Challenge 2. Combinatorial explosion in size of possible world: possible world is a common used model to analyze uncertain data. As it needs enumerate all combinations for records in data set, when the scale of data set is large, it is a great challenge to deal with the combinatorial explosion problem in uncertain data analysis with possible world model.

To figure out the solutions about the challenges in analyzing the large scale data with non-uniform data schema, this paper makes following contributions: (1) proposed the concept "schema uncertainty" formally; (2) proposed couple correlation degree (Cor) to evaluate the existence probabilities for records with "schema uncertainty" and addressed the problems to analyze the data with schema uncertainty; (3) provided B-correlation tree to capture the hierarchical couple correlation degrees for records; (4) proposed approximate algorithms to apply queries on data with schema uncertainty; and (5) provided sufficient experiments to test the effectiveness and efficiency of our methods.

The remaining part of this work is organized as follows: Chapter 2 discussed the recent related work about the uncertain data modeling; Chapter 3 formalized the schema uncertainty and the problems to be solved; Chapter 4 proposed B-correlation tree to capture the schema uncertainty in different levels, and Chapter 4 also proposed accurate and Monte Carlo based algorithms to evaluate queries on data with schema uncertainty and discussed the error rate for the algorithms; Chapter 5 gave sufficient experiments to test the efficiency and effectiveness of modeling uncertain data with different levels of B-correlation tree, analyzed our methods with different parameters and compared their effectiveness and

efficiency both on real medical data and TPC-H benchmark data; Chapter 6 made a conclusion about the whole work and gave our future working directions.

2 Related Works

Uncertain data analysis [1] [2] is based on the probability database theory [3], and it considers a probability dimension to describe the uncertain degree in data for the information lost during the collecting and integration process. BarbarB et al. [4] expanded the uncertain data theory and imported some specific operators to tackle the data with probability dimensions. Chatfield et al. [5] showed the affection of uncertainty in data to the analysis result. Recently, to meet the requirement in enhancing accuracy of analyzing large scale integrated data [6], managing and mining uncertain data are becoming research hotspots for database and data mining fields again. One famous model for uncertain data is possible world model [7]: By using the probabilistic values to describe the uncertainty for records in dataset, possible world model generates all combinations (possible world instances) of records in dataset, and possible world model obtained the analyzing results from uncertain data by summarizing the analyzing results on all possible world instances. Trio [8], Avatar [9] and proTDB [10] are famous uncertain data management system by using possible world model. Khoussainova [11] tried to add constraints into uncertain data to deal with the error imported in the inputting process. Jayram [12] discussed approximate estimation algorithms for the aggregation operations on uncertain data.

Monte Carlo method [13] applies random experiment method [14] to calculate high dimensional integration. Recently, researchers has started to apply Monte Carlo method into database field to solve the efficiency problem in large scale uncertain data query or analysis. MCDB [15] applied Monte Carlo method for uncertain data query based on relational DBMS. It randomly generated possible world instances for uncertain data, and gathered the statistics to estimate the query result. MCDB also designed flexible VG function to evaluate the uncertainty for data records, and it did not directly store the probability dimension for uncertain data. It applied specific statistical distribution to capture the probability dimension for uncertain data. However, the prior statistical distribution it needed lost the objectivity to the problem and thus MCDB has limitations on uncertain data processing. E=MC3 [16] followed the way of MCDB and imported Map-Reduce mechanism to satisfy the enterprise-level distributed efficiency about uncertain data. It proposed a distributed seeding method to generate random variable values follow global uniform distribution under distributed framework. This paper follows the idea of applying Monte Carlo method to analyze uncertain data efficiently, and provides a subject-oriented uncertainty degree evaluating method for uncertain data based on the correlation degree of data records and provided query subject.

3 Problem Definition

Uncertain data results from the integration process of several datasets with different degrees of accuracy limit, data granularity or missing value. Figure 1

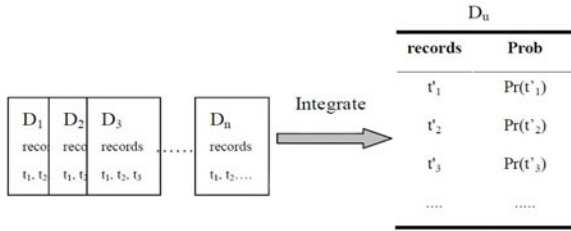


Fig. 1. Generate uncertain data by integration

illustrated the integrated process to generate an uncertain data set. As it is shown in figure 1, datasets from D_1 to D_n are the original datasets, and uncertain dataset D_u is the integration result. To analyze the uncertain data in D_u , its records were given probabilistic values to describe their existence chances in D_u . Possible world model is a basic model to deal with uncertain data and it is composed of several possible world instances. A possible world instance describes a combination of part of records in D_u . Possible world $PW(D_u)$ made up of all possible world instances. For example, as the settings in figure 1, $PW(D_u) = \{\{\emptyset\}, \{t'_1\}, \{t'_2\}, \{t'_3\}, \{t'_1, t'_2\}, \{t'_1, t'_3\}, \{t'_2, t'_3\}, \{t'_1, t'_2, t'_3\}, \dots\}$. And in $PW(D_u)$, the existence probability $\Pr(pw_i)$ for given instance $pw_i \in PW(D_u)$ can be obtained by: $Pr(pw_i) = \prod_{j \in [1, |D_u|]} Pr(t_j)(1 - Pr(t_j))$. By making use of possible

world model, a query on possible world model can describe as a process of: (1) enumerate all possible world instances; (2) evaluate query on these instances; (3) gather the statistics on all instances to generate final query result.

The two challenges to apply this process on uncertain data analyzing are: (1) existence probability evaluation: as the existence probabilistic values actually do not exist in data, we need an efficient and objective way to evaluate the existence probabilistic values of records for it whether appear in possible world instances; (2) combination explosion problem in possible world instances: when the scale of dataset is large, to enumerate all combinations of records is impossible, thus we need an approximate method to estimate the final query result on large-scale uncertain data. To evaluate the existence probabilities for records in uncertain data set, we provide following concepts and definitions.

Definition 1. Given data set $D = \{t_1, t_2, \dots, t_n\}$, for each t_j ($1 \leq j \leq n$), $A_i(t_j)$ refers to the i -th attribute of t_j ; $Schema(t_j) = \{A_1(t_j), A_2(t_j), A_3(t_j), \dots, A_n(t_j)\}$ represents the schema for t_j . If $\exists t_a, t_b \in D$ ($a, b \in [1, |D|]$, $a \neq b$), $Schema(t_a) \neq Schema(t_b)$, then D is a data set with schema uncertainty.

In the practical data integration process: as people needs to integrate heterogeneous data sets, the schema uncertainty exists in the integrated results in many applications prevalently. For example, in our medical analyzing project, doctors wish to find out all defect cases in the integrated birth defect data D which satisfy conditions: (1) from year 1970 to 1990, (2) in the north China areas, and (3) mother age ≤ 30 . As in earlier China, the level of information technology

application is low, the geographic and time information are usually not well recorded. That is only part of records in D can fully match all the attributes related to the analysis subject. This may cause uncertain analyzing results. The more, as the ratio of this partly mismatched data is high in D (over 20% in original data set), we cannot simply remove them from D for this may cause the analysis result bias from the real situation.

Usually, this kind of uncertainty is correlated with the analyzing subject. Through observation we found that queries and records have similar semantic structure: both of them are made up of several couples of "key-value", for example, the question in the former section can be represented by form $Q_1 = \{t|t \in D, t.year=1970, t.year=1971, t.year=1972, \dots, t.area = \text{north China}, t.mother_age = 20, t.mother_age = 21, t.mother_age = 22, \dots, t.mother_age = 30\}$ and for any record $t_i \in D$, t_i can be represented by form $t_i = \{t.year=year_i, t.area = area_i, t.mother_age = age_i, \dots\}$. With the similar semantic structure, to measure the "query-to-record" or "record-to-record" correlation degree can be unified as to calculate the correlation degree between two sets of "key-value" couples. We gave following definition to do this calculation and capture the schema uncertainty.

Definition 2. Let C is a set of key-value couples, $C = \{\langle k_0(C), v_{k_0}(C) \rangle, \langle k_1(C), v_{k_1}(C) \rangle, \dots, \langle k_n(C), v_{k_n}(C) \rangle\}$, for $k_i(C)$ is i -th key (unique in C) and $v_{k_i}(C)$ is the value for $k_i(C)$ in $C(i=0, 1, 2 \dots n)$. $K(C)$ and $V(C)$ are sets of keys and values for C respectively. Then the couple correlation degree (Cor) for $\forall c_1, c_2 \subseteq C$ can be denoted by following equation:

$$Cor(c_1, c_2) = \omega \frac{|K(c_1) \cap K(c_2)|}{|K(c_1) \cup K(c_2)|} \times \prod_{\forall k_i \in K(c_1 \cap c_2)} Sim_c(v_{k_i}(c_1), v_{k_i}(c_2)) \quad (1)$$

Where ω is a parameter to adjust the range for couple correlation degree, and Sim_c represents the content similarity for texts. As it is described in definition 2, to calculate the correlation degree between two key-value couples c_1, c_2 needs to get: (1) correlation degree between $K(c_1)$ and $K(c_2)$; and (2) content similarity between the values for matched keys of c_1 and c_2 . In this paper, we applied n-grams[17] to obtain Sim_c between the contents of two values with same key.

Subject-oriented evaluation and analysis. Follow the setting of definition 2, we suppose $\forall c \in C$ can be any query or record of D in this paper. Thus, we can measure correlation degrees between query and record or two records. In a scene of applying query q on data set D, we use couple correlation degree between q and $\forall t \in D$ as the estimation of the uncertainty degree for t. Thus our methods are based on the analyzing subject. And in the following part, we address the problems to be solved in this paper formally.

Problem 1. Efficient subject oriented uncertainty degree evaluating: Let D is a data set with schema uncertainty, $Q = \{q_1, q_2, q_3, \dots, q_n\}$ is a set of queries, then for $\forall q_i \in Q$ and $\forall t_i \in D$, to efficient calculate the value of $Cor(q_i, t_i)$ and use it as the estimation for uncertain probability of t_i toward q_i .

Problem 2. Efficient uncertain data analysis: Given query $q_i \in Q$, (a) to enumerate possible world $PW(D)$ for D , and then (b) to calculate the summarized query results on $\forall pw \in PW(D)$.

The difficulties to solve problem 1 and 2 are: as the scale of D is large, (1) it's hard to calculate the subject-oriented uncertainty degree efficient for a random record in D for given query, and (2) it's impossible to enumerate all possible world instances for $\forall t_i \in D$ due to the combinatorial explosion problem, thus we need corresponding approximate methods with acceptable accuracy.

To solve the problems, our idea is to: a) evaluate the correlation degree between all records in data set D , and establish hierarchical summary for the records with the correlation degrees between records; b) select records on the proper level of the obtained hierarchical structure to enumerate the possible world instances and evaluate the uncertain degrees for nodes by calculating the couple correlation degree between the given query q and the corresponding node's median record; c) apply q on all enumerated possible world instances to obtain query results and d) gather the statistical results to calculate the final result. Follow these steps, we designed B-correlation tree to summarize the hierarchical correlation degree between records in D and based on B-correlation tree, we designed an efficient Monte Carlo possible world analysis method.

4 Hierarchical Monte Carlo Possible World Analysis

With the description in problem definition, in the application to calculate query or aggregation on the birth defect uncertain data set, we use the correlation degree between query and records in data set to estimate the uncertain probabilistic value. The challenge to apply this method is that if we calculate the correlation degree of all data records in uncertain data set for every query, the system's efficiency would be rather low. And the more, with so many uncertain data records, it is impossible and unnecessary to enumerate all possible world instances for one query. Thus, to release this efficiency problem and keep a proper accuracy for the correlation degree calculation, we design B-correlation tree — a hierarchical summary structure to index records by their correlated relations.

B-correlation tree. Given data set $D = \{t_1, t_2, \dots, t_n\}$, suppose t_c is the record with the max attributes number in D , and we use t_c as the initiate node to generate the B-correlation tree. A m -order B-correlation tree is the extended version of m -order B-plus tree with following conditions: a) every node in m -order B-correlation tree is a m -size set with an unique data record and its correlation degree (key-value couple, Cor in definition 2) with t_c ; b) every node on the same level of B-correlation tree is on a corresponding linked list in order to access nodes on same level of the tree easily.

We established the B-correlation tree by B-plus tree generation algorithm and extended a BFS style method to generate the hierarchical structure for B-correlation tree, and the detail of this algorithm is list in BCorrelationTree-gen.

```

Data: query  $q(*)$ ,  $D$  (all data records);
Result: B-correlation tree  $T$ 
begin
   $T \leftarrow \emptyset$ ;  $t_c \leftarrow$  find record with max attributes number;
  foreach  $t \in D$  do
     $pr \leftarrow Cor(t_c, t)$ ;  $T.insert(t.id, pr)$ ;      /* use B-plus tree style node
    insertion and splitting algorithm for B-correlation tree */
  end
   $T.CreateHierarchicalStruct()$ ;
end
Function createHierarchicalStructure( $T$ );
begin
   $queue_1, queue_2, hstruct \leftarrow \emptyset$ ;  $queue_1.insert(T.root)$ ;
  while  $|queue_1| > 0$  do
     $tmp\_node \leftarrow queue_1.top()$ ;
    if  $tmp\_node.childrenNum = 0$  then return  $hstruct$ ;
    add all children of  $tmp\_node$  into  $queue_2$ ;
    if  $|queue_1| = 0$  then
      add all nodes in  $queue_2$  into a level of  $hstruct$ ;
       $queue_1 \leftarrow queue_2$ ;  $queue_2 \leftarrow$  new queue;
    end
  end
end

```

Algorithm 1. BCorrelationTree-gen (BCTree-gen)

In a B-correlation tree, nodes on each level of the tree keeps the distribution information about D with different accuracy: The higher the level is, the lower the accuracy will be. Meanwhile, the higher the level is, the smaller the partition number of D (nodes number on the level of B-correlation tree) will be, thus, by selecting the nodes in high level of B-correlation tree to enumerate possible world can reduce the size of possible world model, and improve the efficiency of server by sacrificing some accuracy. We proposed theorem 1 to discuss the relation between level and accuracy.

Theorem 1. *Given data set D , $|D|=N$; and given a m -order B-correlation tree T : If to evaluate the distribution of correlation degree for nodes by histogram, the error rate between the distributions of nodes in K -th level of T and the all records in D may satisfy the following equation:*

$$\epsilon_K = \sum_{i \in [0, h]} \frac{n_i^K}{2N} - \sum_{i \in [0, \frac{h}{m^{L-K}}]} \frac{m^{L-K} n_i^{K-1}}{2N^2} \quad (2)$$

Where h is the size for histogram window, and L is the height of T , n_i^K is the number of records which with correlation degree in the corresponding window of a histogram represented the nodes on K -th level of T .

Proof. As m -order B-correlation tree T 's height $L = \text{Log}_m N$, the maximum nodes number at K -th level of T is m^K ($K \in [0, L]$); as after every insertion on T , the records with median correlation degree in nodes will be promoted to the upper node, thus, nodes on $(K-1)$ -th level of T are the m -quantile for the ones on K -th level. Therefore, K -th level nodes partition D into M^K parts (nodes). By partitioning D into $|D|/h$ windows each with h average records, we used a histogram method to analyze the error rate for the distribution records among the nodes in levels of T . To use the nodes in K -th level of B-correlation tree to partition D into m^K part, and $m^K = |D|/h$; thus $h = |D|/m^K$. According to the probability density function of $f(x) = n_x/(2nh)$ (x is the corresponding window, and n_x is the number of records belong to x), the distribution error (ϵ) between K and $(K-1)$ -th level of T is:

$$\epsilon = Pr(x)_K - Pr(x)_{K-1} = \int_{i \in [0, h]} f_i(x) dx - \int_{i \in [0, h/m]} f_i(x) dx \tag{3}$$

Thus:

$$\epsilon = \sum_{i \in [0, h]} \frac{m^K n_i^K}{2N} - \sum_{i \in [0, h/m]} \frac{m^{K-1} n_i^{K-1}}{2N^2} \tag{4}$$

And the error rate between L and K -th level of T is

$$\epsilon_K = \sum_{i \in [0, h]} \frac{n_i^K}{2N} - \sum_{i \in [0, \frac{h}{m^{L-K}}]} \frac{m^{L-K} n_i^{K-1}}{2N^2} \tag{5}$$

Proof End

According to theorem 1, with the higher level of B-correlation tree is selected for uncertain data analysis, the analysis result will prone to the precise result.

Possible World Analysis. With B-correlation tree we can describe dataset D into set of nodes by keeping the data distribution with different granularity, and based on this structure, we proposed a Monte Carlo possible world analysis method to analyze uncertain data efficiently. Before we introduce the Monte Carlo method, we give the naive method for possible world analysis.

Naive Possible World Analysis (naive-evaluate): Given uncertain data set D . (1) Evaluate existence probabilities for each record in D ; (2) Generate possible world model $PW = \{pw_1, pw_2, pw_3, \dots, pw_n\}$ by enumerating all possible world instances ; (3) Gather the statistics: To a given query q , the result of it is applied on $pw_i \in PW$ is $q(pw_i)$. By gathering $q(pw_i)$ s for all $pw_i \in PW$, and calculate the expectation $\frac{\sum_{pw_i \in PW} q(pw_i) \times pr(pw_i)}{|PW|}$, where $pr(pw_i)$ is the existence probability of pw_i in PW , we can get the query result of q on D .

By applying this naive method, for a data set with n records, a complete possible world will need 2^n possible instances. Thus , when n is large (over 1 million), to use naive method to analyze uncertain data is impractical.

Monte Carlo Possible World Analysis. To solve the combination explosion problem in uncertain data analysis, we followed the method of MCDB [15] to use

Monte Carlo style method to enumerate the possible world instances: Based on the structure of B-correlation tree, we first retrieve the node sets at proper level to reduce the size of the obtained data size, and then apply the Monte Carlo style method to evaluate the analysis result on uncertain data. The detail steps are listed in algorithm MonteCarlo-evaluate (MCE).

```

Data: D, B-correlation tree T, level l, query q(*), iterative number N
Result: q(PW)
begin
    Snode ← get set of nodes in l-th level of T;
    count, sum ← 0;
    while count < N do
        foreach nodei ∈ Snode do
            tmi ← find the record with median Cor value in the i-th node;
            Vq[i] ← Cor(tmi, q);
        end
        enumerate all bits for Vpresent randomly;
        δ(PW) = Vpresent × VqT × [Cor(tm0, q), Cor(tm1, q) ... Cor(tm|Snode|, q)]T;
        sum += Q(δ(PW) × |Snode|) count++;
    end
    q(PW) = E[Q(δ(PW))] ← sum/N;
end

```

Algorithm 2. MonteCarlo-evaluate (MCE)

Where q(PW) refers to the result of query or aggregation analysis on possible world PW, V_{present} = [a₁, a₂, ..., a_n] is a binary vector refers to the existed status of a node in one possible world instance (a_i = 1 means the i-th node exists in this possible world instance and a_i = 0 means it does not existed in this instance), and V_q = {s₁, s₂, ..., s_n} is a vector of score for query q on each node.

As it is shown in algorithm 2, algorithm MCE is an iterative algorithm. Thus, we provided following theorem to analyze the relationship between the iterative number and the accuracy of the algorithm result for MCE.

Theorem 2. Given uncertain data set D, a specific query q. If we apply naive-evaluate and MCE algorithms on D to get the query result for q respectively, then: (1) The result of MCE will converge to the result of naive-evaluate with the N increasing; (2) Denote the precise result of q on D is f(x), and when f(x) < 0.5, the probability that the result error of MCE on D will smaller than ε is δ after the algorithm iteratively run R * ln(2/δ)/ε² times, where R is constant.

Proof: As the former descriptions, V_{present} refers to the existed status of each node for corresponding possible world instance, suppose the present frequency of node n_i in B-correlation tree is f_i, then according to "Bernoulli great number law", $\lim_{N \rightarrow \infty} p\{|f_i/N - p_i| < \epsilon\} = 1$, where N refers to the iterative times. Thus, when N is prone to a big enough number, the frequency for one node presented

in enumerated possible world instance is close to its existed probability p_i , and thus, the result of MCE will converge to the result of naive-evaluate method **(1) proof ends**; According to the great number law [18] and the lemma in literature [19]: let p is the mean of random variable and when $p < 0.5$, to all the normalized i.i.d. random variable $\{x_i\}$ for same distribution in range $[0, 1]$, the following inequality will be tenable.

$$Pr[(x_1 + x_2 + \dots x_t)/t - p | - p > \epsilon p] < 2exp[-2\epsilon^2 tp/9(1-p)] \quad (6)$$

Where ϵ is a constant and $\epsilon \in (0,1)$, t is the number of iterative times (or the number of random experiments). As the result of MCE, denote as $f(x)'$, is the expectation of the results on each instance of possible world, and each result are random variable by following same distribution and independent to another. Suppose the accurate result for naive-evaluate applied on D is $f(x)$, and according to equation 2 and former descriptions, we obtained following inequality:

$$Pr[(f(x)' - f(x) > \epsilon f(x)] < 2exp[-2\epsilon^2 t \cdot f(x)/9(1-f(x))] \quad (7)$$

As $Pr[(f(x)' - f(x) > \epsilon f(x)]$ is a probability constant can be obtained in experiments, we suppose this probability is δ , and thus, the upper bound of δ is $\delta = 2exp[-2\epsilon^2 tp/9(1-f(x))]$. As $f(x)$ can also obtain from experiments. By writing $9(1-f(x))/(2f(x))$ as R , $t = R \cdot \ln(2/\delta)/\epsilon^2$, **(2) proof ends**.

As theorem 2 can be used to estimate the iterative number of random experiments, we tested it in the experiments and applied it in prototype system to predict the iterative number for the MCE with given accuracy.

5 Experiment Discusses and Applications

Datasets and Configuration: We tested our methods both on real birth defect dataset (BD) from CBDMC and TPC-H [20] dataset. TPC-H is a commercial test standard for query efficiency, we used its provided synthetic generator to generate 6,001,215 rows data for our experiments. Table 1 shows a summary of the datasets used in this paper. We implemented the prototype by JAVA (currently, the prototype system has implemented basic query and COUNT(*) aggregate operator), and used a PC with Pentium Dual 1.80GHZ CPU and 2GB RAM as analysis server. All data used in the experiments are stored in MySQL.

The queries used in the experiments are: (1) Queries for BD data: $Q_1 = \text{"SELECT COUNT (*) FROM BrithDefects WHERE year=1980s AND region=north_china AND mother_age>18 AND mother_age<30"};$ $Q_2 = \text{"SELECT COUNT (*) FROM BrithDefects WHERE year=1988 AND mother_age\geq 25 AND mother_age\leq 35 AND mother_family_disease = non AND father_family_disease = non AND region=north_China"};$ $Q_3 = \text{"SELECT COUNT (*) FROM BrithDefects WHERE AND mother_age\geq 25 AND mother_age\leq 35 AND mother_family_disease = non AND father_family_disease = non AND region=north_China AND macrostomia = true AND harelip = normal"}.$ (2) Queries for TPC-H data: $Q'_1 = \text{"SELECT COUNT(*) FROM$

Table 1. A summary of data sets

	Birth defect(BD)	TPC-H data
rows	106,476	6,001,215
Max.attribute number	245	500

lineitem WHERE l_shipdate > 1998-12-01 AND L_LINENUMBER = 500 AND L_DISCOUNT = 0.73 AND L_EXTENDEDPRICE = 1,000”.

Effectiveness Experiments. Our prototype system generated a 8 levels B-correlation tree by setting order m=10. As the nodes in B-correlation tree contains the correlation degree between each record and t_c , we tested the affection of the initial node t_c selection. And we found in experiment that: with different t_c our method generated similar B-correlation trees with 8 levels, thus the affection of the initial node for the distributions of B-correlation tree node on different levels can nearly be ignored.

We applied $Q_1 \sim Q_3$ on BD dataset for 5 times and use the average results to analyze the relationship between relative parameters and the query results. To obtain the summarized query result of possible world instances, we compared direct mean (non-weighted) and the mean with existence probability weighted (weighted) of results on all possible world instances. That is, suppose result(pw_i) is the query result on pw_i , and weighted result = $\sum_{pw_i \in PW} Pr(pw_i) \times result(pw_i)$, and non-weighted $\sum_{pw_i \in PW} result(pw_i)$ (i=0,1,2,3,...). Figure 2 showed the relation between level and query result of Q_1 (both weighted and non-weighted) on BD data with iterative times scale = 1000.

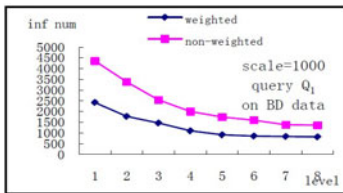


Fig. 2. Relation of level &. result

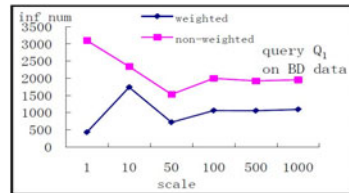


Fig. 3. Relation of scale &. result (level=4)

By the concept of B-correlation tree, to select nodes on level 8 will contain all records in BD (the height of this B-correlation tree on BD dataset is 8). As it is shown in figure 2, both weighted and non-weighted query result are prone to the accuracy results (the results of level 8). We also compared the affection of iterative times to the query result, and figure 3 shows the query result on BD data set with different iterative times (scale). As it is shown in figure 3, to apply MCE on data with fixed level of B-correlation tree (level=4 in this experiment), the query result will converge to fixed value with the iterative times (scale)

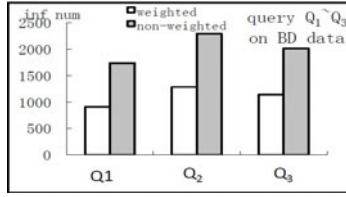


Fig. 4. Comparison of $Q_1 \sim Q_3$ (level=5, scale=1000)

increasing. The result in figure 3 satisfies theorem 2 that the needed iterative times scale =1000 ($1000=80 \cdot R$ with $R \approx 12.5$) can make sure "the probability δ of error rate " $\epsilon < 0.01$ " bigger than 0.9.

To test the correctness of MCE, we compared all results of Q_1, Q_2, Q_3 on BD data with level 5 and scale =1000, and the result is shown in figure 4. As it is shown in figure 4, during January to December of the year 1988, the birth defect number satisfies the conditions of Q_2 is the most of the three queries, that means anencephalus is the main birth defect in that year. According to the medical survey in China [21] the disease incidence rate of anencephalus is indeed higher than other birth defects in China during the year 1980 to 2000.

Efficiency Experiments. In the rest part of experiments, we test the efficiency of the prototype system with the provided algorithms on TPC-H data. To save the memory space cost, we applied pages method in the prototype system to read 200,000 records every time for a page and generate B-correlation tree ($n=50$) incrementally. And the figure 5 and 6 are the time and space cost for our prototype system respectively. As it is shown in figure 5, the time cost for our prototype to establish B-correlation tree is linear with the number of records, and t. The total time for prototype to complete a B-correlation tree on 6,001,215 records is about 30 minutes. And the space cost for prototype system to complete B-correlation tree is also linear with the number of records, and in this experiment, the final space cost after completing the whole TPC-H records is 453,324 KB, and to serialize the obtained B-correlation tree to hard disk cost another 138,607 KB space. Finally, we obtained a 5-level B-correlation tree with each node contains 50 records and their correlation degree value.

Scalability for MCE algorithm on TPC-H data. We analyzed the MCE by selecting all 5 levels (0~4) respectively of the obtained B-correlation tree.

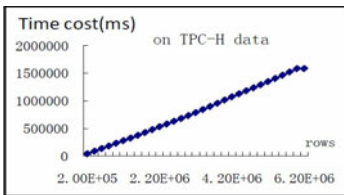


Fig. 5. Time cost for BCTree-gen

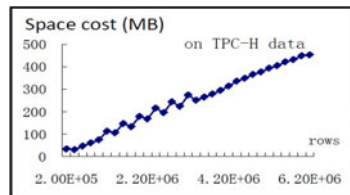


Fig. 6. Space cost for BCTree-gen

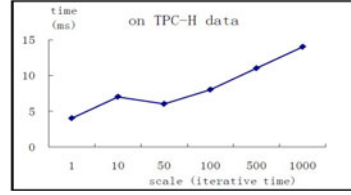
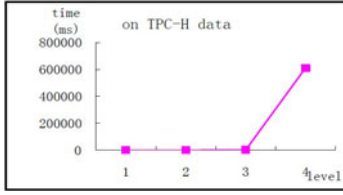


Fig. 7. Time cost with different levels **Fig. 8.** Time cost for different iterative time

As it is shown in figure 7, the time costs of level 0~3 can be accepted within 2,936 milliseconds, the accurate query on all data cost 609,681 milliseconds (4-th level). We also test the time cost for different iterative time of the prototype system: figure 8 shows the result by applying Q'_1 on nodes of level 3, and the time cost is almost linear with iterative time.

6 Conclusion

In this paper, we proposed the concept of schema uncertainty in the medical data mining project for CBDMC, and provided methods to do efficient probability evaluating and analysis for uncertain data. The main contributions of this work are: (1) proposed the concept of schema uncertainty to describe the uncertainty in data caused by integrating heterogeneous data sets, and proposed couple correlation degree to evaluate the subject-oriented uncertainty between records and query subject; (2) proposed B-correlation tree to hierarchical model the uncertain data with their couple correlation degree, and discussed the distribution affection for nodes in different levels of B-correlation tree; (3) provided a Monte Carlo style method to implement the efficient uncertain data analysis or mining, and gave discussion about the relation between iterative times and result accuracy; (4) provided sufficient experiment to test the effectiveness and efficiency for our method both on medical data from CBDMC and TPC-H benchmark. Our method is a good way to analyze uncertain data from integrated heterogeneous data sets and it can evaluate the uncertainty degree for data based on the analyzing subject and it is a promising direction to solve the large scale uncertain data analyzing problem with little prior settings. The future working directions for our work are: to extend the prototype system to provide more aggregation operators, and to try to implement more data mining methods in prototype system in order to make it more efficient and more usable to our medical data analyzing task or other uncertain data mining tasks.

References

1. Aggarwal, C.C.: Managing and Mining Uncertain Data. In: Advances In Database Systems (2009)
2. Sarma, A.D., Benjelloun, O., Halevy, A., Widom, J.: Working Models for Uncertain Data. In: ICDE 2006 (2006)

3. Cavallo, R., Pittarelli, M.: The Theory of Probabilistic Databases. In: Proceedings of the 13th VLDB Conference, Brighton (1987)
4. BarbarB, D., Garcia-Molina, H., Porter, D.: The management of probabilistic data. *IEEE Transactions on Knowledge and Data Engineering* 4(5), 487–502 (1992)
5. Chatfield, C.: Model Uncertainty, Data Mining and Statistical Inference. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 158(3), 419–466 (1995)
6. Dong, X.L., Halevy, A., Yu, C.: Data integration with uncertainty. *The VLDB Journal* 18(2), 469–500 (2009)
7. Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic Frequent Itemset Mining in Uncertain Databases. In: Proc. 15th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2009), Paris, France (2009)
8. <http://infolab.stanford.edu/trio>
9. <http://www.almaden.ibm.com/cs/projects/avatar>
10. <http://www.math.ups.edu/~anierman/umich/prodb>
11. Khoussainova, N., Balazinska, M., Suciu, D.: Towards Correcting Input Data Errors Probabilistically Using Integrity Constraints. In: *MobiDE 2006*, June 25 (2006)
12. Jayram, T.S., McGregor, A.: Estimating Statistical Aggregates on Probabilistic Data Streams. In: *PODS 2007*, June 11-14 (2007)
13. Metropolis, N., Ulam, S.: The Monte Carlo Method. *Journal of the American Statistical Association* 44(247), 335–341 (1949)
14. Stigler, S.M.: A Historical View of Statistical Concepts in Psychology and Educational Research. *American Journal of Education* 101(1), 60–70 (1992)
15. Jampani, R., Xu, F., Wu, M., Perez, L.L., Jermaine, C., Haas, P.J.: MCDB: a monte carlo approach to managing uncertain data. In: Proceedings of the 2008 ACM SIGMOD (2008)
16. Xu, F., Beyer, K., Ercegovac, V., Haas, P.J., Shekita, E.J.: E = MC3: managing uncertain enterprise data in a cluster-computing environment. In: Proceedings of the 2009 ACM SIGMOD (2009)
17. Li, Y., Algarni, A., Zhong, N.: Mining positive and negative patterns for relevance feature discovery. In: *KDD 2010 Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2010)
18. Renyi, A.: *Probability theory*. NorthHolland, Amsterdam (1970)
19. Karp, R., Luby, M.: Monte-Carlo Algorithms for Enumeration and Reliability Problems. In: 24th STOC, pp. 56–64 (1983)
20. <http://www.tpc.org/tpch/>
21. Yang, H., Cai, H.: Clinicopatholog analysis on 46 inborn anencephaluses. *Chinese Journal of Birth Health and Heredity* (2008)

An Empirical Evaluation of Bagging with Different Algorithms on Imbalanced Data

Guohua Liang and Chengqi Zhang

The Centre for Quantum Computation & Intelligent Systems, FEIT,
University of Technology, Sydney NSW 2007 Australia
{gliang,chengqi}@it.uts.edu.au

Abstract. This study investigates the effectiveness of bagging with respect to different learning algorithms on Imbalanced data-sets. The purpose of this research is to investigate the performance of bagging based on two unique approaches: (1) classify base learners with respect to 12 different learning algorithms in general terms, and (2) evaluate the performance of bagging predictors on data with imbalanced class distributions. The former approach develops a method to categorize base learners by using two-dimensional robustness and stability decomposition on 48 benchmark data-sets; while the latter approach investigates the performance of bagging predictors by using evaluation metrics, True Positive Rate (*TPR*), Geometric mean (*G-mean*) for the accuracy on the majority and minority classes, and the Receiver Operating Characteristic (*ROC*) curve on 12 imbalanced data-sets. Our studies assert that both stability and robustness are important factors for building high performance bagging predictors on data with imbalanced class distributions. The experimental results demonstrated that PART and Multi-layer Proceptron (MLP) are the learning algorithms with the best bagging performance on 12 imbalanced data-sets. Moreover, only four out of 12 bagging predictors are statistically superior to single learners based on both *G-mean* and *TPR* evaluation metrics over 12 imbalanced data-sets.

Keywords: classification, bagging, imbalanced data, and ROC curve.

1 Introduction

Bagging [1] utilizes “bootstraps samples” to build a set of individual classifiers to make predictions and the final decision is determined by a majority vote of the predictions of the individual classifiers in the ensemble. Breiman pointed out that instability is an important factor for bagging to improve accuracy by reducing variance [1]. Bagging is widely regarded as a variance-reduction technique, so it is mostly applied to unstable, high variance algorithms to improve predictive accuracy. Our studies demonstrate that both stability and robustness are key factors for bagging to achieve a high performance prediction model.

Learning from imbalanced data has become one of the crucial issues in Machine Learning and Data Mining communities, due to the increasing number of real world

applications involving extremely skewed class distribution and/or unequal cost of different mis-classification errors in minority and majority classes, e.g., credit card fraud detection [2] and detection of oil spills in satellite radar images [3]. In these situations, the minority class has a notably higher cost than the majority class, and it is required to achieve high accuracy in the minority class, however, the predictive accuracy or error rate is an ineffective measure for evaluating the performance of minority class [4, 5].

The effectiveness of the bagging has been empirically evaluated in the literature [5-10]. Most of these previous studies evaluated the performance of bagging by associating it with other ensemble learning methods based on one or a few learning algorithms [7-10], such as C4.5 decision trees and Neural Networks by using an accuracy or error rate as a performance measure, which is considered as an inappropriate evaluation measure for extremely skewed class distribution and/or unequal costs of mis-classification errors [4] [11]. In addition, most did not conduct statistical comparisons to draw their conclusions in the context of the imbalanced learning.

Our previous studies have carried out statistical comparisons to investigate the performance of bagging intensively across a rich set of base learners in general terms [6] and in imbalanced situations [5]. In the earlier study [6], instance bias/variance decompositions 0/1 loss are used as performance measures to classify the base learners into different categories, and the error rate is used as a performance measure to investigate the performance of bagging with respect to different learning algorithms in general terms. As error rate is not an appropriate performance measure for imbalanced learning, the experimental results cannot be applied to imbalanced situations. In the later study [5], we investigated the Area Under the ROC Curve (*AUC*) performance of bagging with respect to different levels of class distribution. *AUC* performance measure was used as an evaluation metric for this study, because *AUC* has been considered as an alternative measure for comparing the performance of the classifiers across the entire range of class distributions and error costs [12-14]. Due to limited space, we could not provide any *ROC* graphs and other evaluation metrics in the paper.

Our current research investigates the effectiveness of bagging with respect to different learning algorithms on imbalanced data-sets based on other evaluation metrics, such as *TPR*, *G-mean* and *ROC curves* to answer practical questions such as: which bagging predictor has the best performance in imbalanced applications, and whether a bagging predictor is superior to single learners in such a situation? Answering these questions poses the following research challenges: (1) how to classify the base learners into different categories based on error rate/variance in general terms, and (2) how to conduct a valid and rigorous study to evaluate multiple algorithms on imbalanced data-sets [15].

Our main contribution is twofold: (1) provide a clear picture of the two dimensional robustness and stability decomposition to classify the base learners into different categories based on error rate/variance performance measures in general terms, and (2) conduct statistical comparisons to investigate the performance of bagging predictors based on evaluation metrics, *G-mean*, *TPR* and *ROC* on imbalanced data-sets.

The paper is organized as follows. Section 2 outlines the detail of the designed framework. Section 3 presents the evaluation of the base learners and Section 4

presents evaluation metrics. Section 5 provides the experimental setting and Section 6 presents the experimental results analysis. Section 7 concludes by summarizing the significant results of the paper.

2 Designed Framework

The designed framework is presented in Fig. 1, and the evaluation is divided into three tasks: (1) perform robustness and stability decomposition to classify base learners based on error rate/variance as performance measures in general terms, which provides a clear picture of categorized classifiers, (2) compare bagging predictors with single learners: (a) the Wilcoxon signed-ranks test is used to compare two learners to determine when bagging outperforms each single learner in imbalanced situations based on two evaluation metrics, *G-mean* and *TPR*, and (b) the comparison of *ROC* curves is used as an evaluation metric to determine whether bagging MLP is superior to a single learner MLP with respect to different levels of class distribution on the imbalanced data-sets, (3) the Friedman test with the corresponding Post-hoc Nemenyi test is used to compare multiple learners to determine which bagging predictors have the best performance based on two evaluation metrics, *G-mean* and *TPR*.

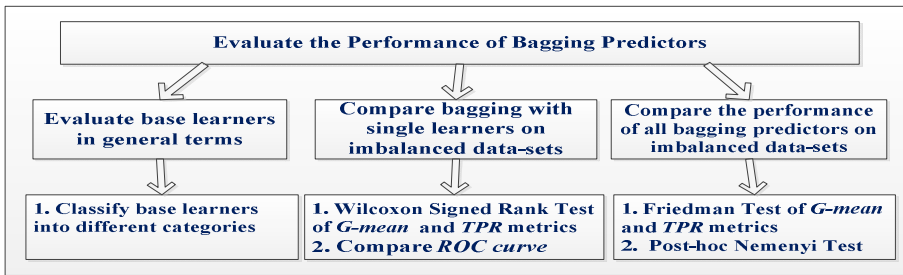


Fig. 1. Designed Framework

2.1 Different Levels of Sample Distributions to Form a ROC

In the literature, sampling techniques have been investigated in the context of imbalanced learning, and the authors [16] respect that sampling will produce the same effect as moving the decision threshold or adjusting the cost matrix. Their experimental results showed that the over- and under- sampling procedures produced ROC curves almost identical to those produced by varying the decision threshold of Naïve Bayes. In addition, ROC curve has been considered as a performance measure for comparing the performance of the classifiers across the entire range of class distributions and error costs [13, 14]. We utilize the *ROC* graphs to compare the performance of the bagging MLP and a single learner MLP with respect to different levels of class distribution on each of the imbalanced data-sets.

We adopt generic sampling techniques to change the class distribution of the data. This investigation utilizes the under-sampling technique to alter each original imbalanced data-set into *nine* new data-sets with *nine* different imbalanced levels of class distribution to investigate the performance of bagging predictors over multiple imbalanced data-sets, e.g., to alter each original imbalanced data-set, D with sample size M into *nine* new data-sets, $D_1, D_2 \dots D_9$ with new sample size $M_1, M_2 \dots M_9$, respectively.

We regard the entire minority class samples as a positive class, then randomly select the majority class without a replacement as a negative class e.g.,

- Select all the minority class as a positive class (sample size P)
 $P = 10\% M_1 = 20\% M_2 = \dots = 90\% M_9$, respectively.
- Randomly select majority class as a negative class (sample size $N_1, N_2 \dots N_9$), so $N_1 = 90\% M_1; N_2 = 80\% M_2 \dots N_9 = 10\% M_9$, respectively.
- *Nine* new data-sets, $D_1, D_2 \dots D_9$ (with new sample size $M_1, M_2 \dots M_9$, respectively) are formed by $(P + N_1), (P + N_2) \dots (P + N_9)$. Therefore, each original imbalanced data-set, D is altered into *nine* different levels of sample distributions.

We perform 10-trial 10-fold cross-validation on each of new data-sets, $D_1, D_2 \dots D_9$, respectively, so that the test-set has the same distributions as the training-set. For example, for a learning algorithm MLP on each original data-set, two *ROC curves* are formed; one for bagging MLP, and the other for single learner MLP, by obtaining *nine* pairs of (FPR, TPR) , respectively. Therefore, each *ROC curve* represents the performance of bagging or a single learner at nine different levels of class distribution.

2.2 Statistical Test

Aiming to conduct rigorous and valid comparisons of the performance of bagging predictors, non-parametric tests were employed for the statistical comparison of learners: (1) Wilcoxon signed-rank test, and (2) Friedman test with the corresponding post-hoc Nemenyi test as follows:

(1) The Wilcoxon Signed-rank test for comparison of two learners over multiple data-sets, e.g., comparison of the performance of bagging SVM and a single learner SVM at nine different levels of class distribution to determine whether the bagging SVM is superior to the single learner SVM on imbalanced data-sets.

(2) The Friedman test with the corresponding Post-hoc Nemenyi test for comparison of multiple learners over multiple imbalanced data-sets, e.g., comparison performance of bagging predictors at nine different levels of class distribution one against another on imbalanced data-sets.

For more details on how to use the Wilcoxon Signed-rank test comparing two learners, and how to use the Friedman test with the corresponding Post-hoc Nemenyi test comparing multiple learners to obtain mean ranks, to identify which mean ranks are significantly different to each other, and how to calculate the critical difference please see the literature [5] [15].

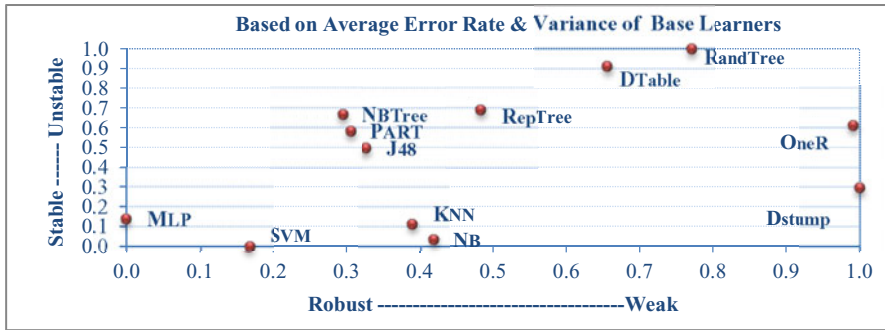


Fig. 2. Two-dimensional robustness and stability decomposition, where the x-axis indicates the ascending ranking orders of the estimated error rate from robust to weak, and the y-axis indicates the ascending ranking orders of the variance from stable to unstable

3 Classify Base Learners

Aiming at investigating the performance of bagging with respect to different learning algorithms, we design a two-dimensional robustness and stability decomposition to characterize base learners based on the estimated average error rate/variance evaluation measures.

Fig. 2 demonstrates the Robustness vs. Stability decomposition based on the estimated error rate and variance as evaluation measures to categorize base learners.

- Firstly, we rank all the base learners on each data-set according to the estimated error rate and variance of each base learner, respectively. If there is a tie, their average value will be the ranks of the base learners.
- Secondly, the average ranks of the estimated error rate and variance were obtained over 48 data-sets.
- Thirdly, normalized ascending rank orders of the estimated error rate and variance were calculated for two-dimensional plotting, where the x-axis indicates the normalized ranking order of the estimated error rate, and the y-axis indicates the normalized ranking order of the estimated variance.
- Finally, we considered the normalized ranking order of the estimated error rate and variance as the robustness and stability of base learners, respectively.
- For example, MLP and SVM with a smaller value of robustness denote a more robust learner, while NB and SVM with a smaller value of stability denote more stable learners.

In Fig. 2 we observe that the group of learners, OneR and DStump are the weakest base learners; the group of learners, MLP, SVM, NBTree and PART are the most robust learners; the group of learners, RandTree and DTable are the most unstable learners; and the group of learners, SVM, NB, KNN and MLP are the most stable learners. MLP and SVM, both having relatively lower variance, are similar to, and have more robustness than KNN and NB, respectively. Therefore, SVM and MLP should be considered as more stable learners, similar to NB and KNN.

4 Evaluation Metrics

Accuracy or error rate is commonly used measure for evaluating the performance of a classifier in general terms. However, it is an ineffective metric for measuring the performance of prediction model on extremely imbalanced data-sets. As in real world applications, the proportion of the minority class is much smaller than the whole population; the minority class is the class which users are interested in; normally a high prediction accuracy is required in a minority class; however, accuracy or error rate has a limitation to evaluate the performance of a classifier on a minority class [17]. We therefore select *TPR*, *G-mean* and *ROC curve* as evaluation metrics for this empirical study.

Table 1 presents the confusion matrix for a binary classification problem. The columns represent the Predicted Positives and Negatives in each class, respectively; the rows represent the Actual Positives (*P*) and Negatives (*N*) in each class, respectively. True Positive (*TP*) refers to the number of actual positive instances that are correctly predicted as the true positive class; False Negative (*FN*) refers to the number of the actual positive instances that are incorrectly predicted as the negative class; False Positive (*FP*) refers to the number of actual negative instances that are incorrectly predicted as the positive class; True Negative (*TN*) refers to the number of actual negative instances that are correctly predicted as the negative instances.

Table 1. Confusion matrix for a binary classification problem

	Predicted Positives	Predicted Negatives
Actual Positives (<i>P</i>)	True Positive (<i>TP</i>)	False Negative (<i>FN</i>)
Actual Negatives (<i>N</i>)	False Positive (<i>FP</i>)	True Negative (<i>TN</i>)

Table 2. *TPR*, *TNR* and *G-mean*

$TPR = \frac{TP}{TP+FN}$	$FPR = \frac{FP}{FP+TN}$
$TNR = \frac{TN}{TN+FP}$	$FNR = \frac{FN}{FN+TP}$
$G - mean = \sqrt{TPR * TNR}$	

Table 2 presents the formulas of True Positive Rate (*TPR*), False Positive Rate (*FPR*), True Negative Rate (*TNR*), and False Negative Rate (*FNR*) in the first and second rows and the formula of *G-mean* of the accuracy on the majority and the minority classes in the last row. *G-mean* [18, 19] is recommended as a performance measure to compare different algorithms by monitoring both the accuracy rates of the majority and the minority classes.

A *ROC curve* has been considered as a useful performance metric for evaluating and comparing the performance of learning algorithms, as the *ROC curve* has special

properties for comparing the performance of the classifiers across the entire range of class distributions and error costs [13, 14].

A ROC graph [18, 20] is a two-dimensional plot, where x-axis denotes false positive rate (*FPR*) of a classifier, and y-axis denotes true positive rate (*TPR*) of a classifier. In the *ROC* plot, the upper left point (0,1) is the most desired point, known as “ROC Heaven” presenting 100% true positive and zero false positives, while the point (1,0) is the least desired point, called “ROC Hell”. In the *ROC* space, when we compare the performance of classifiers at different points, one point closer to the “ROC Heaven” is better than one that is further away.

5 Experimental Setting

We implement the bagging predictor in Java platform and perform 10-trial 10-fold cross-validations to evaluate the performance of bagging and single learners. The 12 learning algorithms are employed from WEKA implementation with default parameter settings in this empirical study [21].

In order to reduce uncertainty and obtain reliable experimental results, all the evaluations are assessed under the same test conditions e.g., all bagging predictors use the same randomly selected bootstrap samples with replacements in each fold of 10-trial 10-folds cross-validation on each data-set.

We classify the base learners into different categories, the experimental results based on 48 data-sets in general terms. Moreover, we investigate the effectiveness of bagging with respect to different levels of class distribution based on 12 imbalanced binary class data-sets.

We adopt an under-sampling method to alter each original data-set into nine different levels of class distribution without replacements. We use the nine pairs of (*FPR*, *TPR*) to form a *ROC curve* for each prediction model on an imbalanced data-set; therefore, each *ROC* graph represents the average performance of a prediction model at nine different levels of class distribution.

5.1 Data-Sets

The selection of the 48 data-sets covers the number of instances, which varied from small to large (up to 20,000), the number of attributes, which varied from 5 to 70, the number of classes, which varied from binary classes to multiple classes (up to 29), and the frequency of the classes included balanced data-sets and extremely unbalanced data-sets.

Table 3 presents a summary of the characteristics of the 48 data-sets, which have been collected from the UCI Machine Learning Repository [22]. The first and the fourth columns indicate the ID number and the name of the data-sets. The second and the fifth columns present the information about the data itself which includes the number of attributes (excluding the class) and the number of instances in each data-set, respectively. The third and the last columns present the information about the classes and the number of classes in each data-set.

Table 3. 48 data-sets characteristics

Data-sets		Info. data			Data-sets		Info. Data		
ID	Name	Attr.	Inst	Cls	ID	Name	Attr.	Inst.	Cls
1	abalone	9	4177	29	25	lymph	19	148	4
2	anneal	39	898	6	26	monks1	7	432	2
3	audiolog	70	226	24	27	monks2	7	432	2
4	auto-m	8	398	3	28	monks3	7	432	2
5	balance	5	625	3	29	mushroom	23	8124	2
6	breastc	10	286	2	30	pima	9	768	2
7	bupa(liv	7	345	2	31	segment	20	2310	7
8	car	7	1728	4	32	sick	30	3772	2
9	cmc	10	1473	3	33	sonar	61	208	2
10	colic(ho	23	368	2	34	soybean	36	683	19
11	crx	16	690	2	35	spambase	58	4601	2
12	crx-g(ge	21	1000	2	36	splice	61	3190	3
13	diabetes	9	768	2	37	staheart	14	270	2
14	ecoli	8	336	8	38	ta	6	151	3
15	glass	10	214	7	39	tic-tac-	10	958	2
16	hayes	5	132	3	40	tumor	18	339	22
17	heart-c	14	303	5	41	vehicle	19	846	4
18	heart-h	14	294	5	42	vowel	14	990	11
19	ionosphe	35	351	2	43	waveform	41	5000	3
20	iris	5	150	3	44	wbreastc	10	699	2
21	kr-vs-kp	37	3196	2	45	wdbc	31	569	2
22	labor	17	57	2	46	wine	14	178	3
23	led	25	1000	10	47	yeast	9	1484	10
24	letter	17	200000	26	48	zoo	17	101	7

Table 4. Imbalanced Data-Sets characteristics

Data-sets		Information Data		Class Data	
Index	Name	Attribute	Instances	Frequency	classes
1	breastc	10	286	201, 85	2
2	bupa	7	345	145, 200	2
3	crx	16	690	307,383	2
4	Crx-g	21	1000	700,300	2
5	diabetes	9	768	500, 268	2
6	ionosphere	35	351	126,225	2
7	labour	17	57	20,37	2
8	stalogheart	14	270	120, 150	2
9	sick	30	3772	3541, 231	2
10	sonar	61	208	97,111	2
11	Tic-tac-toe	10	958	626,332	2
12	WDBC	31	569	212,357	2

A summary of the characteristics of the 12 imbalanced data-sets is displayed in Table4. The data-sets were employed by different criteria, such as the number of instances, the number of attributes, the number of classes and the frequency of each class.

5.2 Selection of Base Learners

The 12 single learners which were employed from WEKA in this study are as follows: Naïve Bayes (NB) is a Naïve Bayes learner; K-nearest-neighbors (KNN) is a IBK lazy learner; SVM is SMO of Support Vector Machines; Multi-layer Perceptron

(MLP) is a neural network learner; PART, Decision Table (DTable), and OneR are rule learners; C4.5 Decision Tree (J48), Decision Stump (DStump), Random Tree (RandTree), RepTree and Naïve Bayes Trees (NBTree) are tree family learners.

6 Experimental Results Analysis

This section presents the experimental results analysis including three sub-sections as follows: A. compare bagging with each of the single learners using the Wilcoxon Signed Rank Tests, B. compare bagging MLP with a single learner MLP using ROC curve, and C. compare bagging predictors against one another using Friedman with Post-hoc Nemenyi tests.

6.1 Compare the Performance of Bagging and Single Learners

This subsection compares the performance of bagging and single learners based on The Wilcoxon Signed-Rank Tests by using two different evaluation metrics, *G-mean* and *TPR*.

The Wilcoxon Signed-Rank Test is used to determine whether there really is an improvement between two learners, e.g., bagging NB and single learner NB. The Null Hypothesis is that the median of differences between Bagging and each single learner equals 0.

Rule: Reject the Null Hypothesis if the *p*-value Test Statistic *W* is less than $\alpha=.05$ at the 95% confidence level of significance.

Table 5. Wilcoxon Signed-Rank Test based on *G-mean* Metric. The significance level is .05.

Bagging v. Each Single Learner on Wilcoxon Signed-Rank Test based on <i>G-mean</i>						
Learners	J48	RepTree	RandTree	NB	SVM	Dstump
p-values	.005	.015	.008	.610	.131	1.000
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	.037	.814	.005	.657	.136	.019

Table 6. Wilcoxon Signed-Rank Test based on *TPR* metric. The significance level is .05.

Bagging v. Each Single Learner on Wilcoxon Signed-Rank Test based on <i>TPR</i>						
Learners	J48	RepTree	RandTree	NB	SVM	DStump
p-values	.015	.002	.005	.906	.722	.263
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	.110	.272	.006	.575	.002	.136

Tables 5 and 6 present the summarized results of the Wilcoxon signed-rank test for the comparisons of the performance of bagging and single learners. If a calculated *p*-value is greater than α value, 0.05, then the *p*-values are highlighted in red and we accept the Null Hypothesis; for example KNN, NB, SVM, DStump, DTable and NBTree in Table 5. For all other cases, we reject the Null Hypothesis.

Table 5 experimental results indicate that only 50% bagging predictors perform better than the single learners, including J48, RepTree, RandTree, OneR, PART and MLP learners, while using *G-mean* as an evaluation metric on 12 imbalanced data-sets,. Table 6 experimental results indicate that less than 50% bagging predictors perform better than the single learners, including J48, RepTree, RandTree, PART and NBTree learners, while using *TPR* as an evaluation metric on 12 imbalanced data-sets,.

Overall, based on both *G-mean* and *TPR* evaluation metrics, only four out of 12 bagging predictors are statistically superior to single learners on 12 imbalanced data-sets. The four bagging predictors are tree family learners, J48, RepTree and RandTree, and rule learner, PART.

6.2 Compare the ROC Curves of Bagging MLP with a Single Learner MLP

This sub-section compares the *ROC curves* between bagging MLP and a single learner MLP to further examine whether bagging is superior to a single learner MLP on 12 imbalanced data-sets.

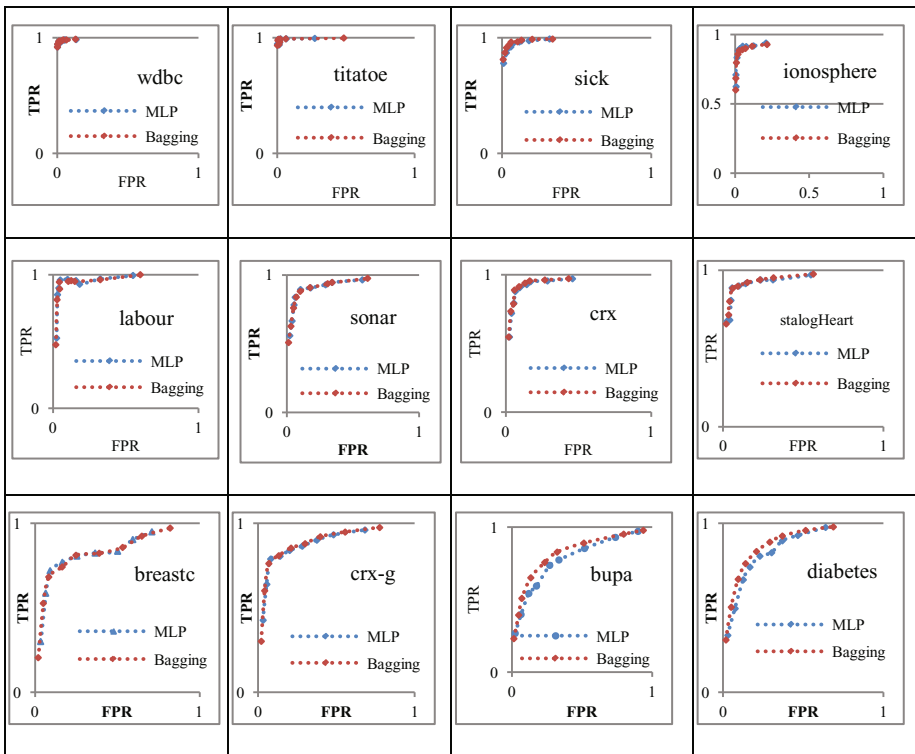


Fig. 3. The group of comparisons of *ROC curves* between a bagging MLP and a single learner MLP on 12 imbalanced data-sets, where the x-axis denotes *FPR*, and the y-axis denotes *TPR* for each sub-Figures

Fig. 3 presents the group of comparisons of *ROC curves* between the bagging MLP and the single learner MLP on 12 imbalanced data-sets; each sub-figure presents two ROC curves, one for Bagging and the other for a single learner on each data-set; and each ROC curve is formed by nine pairs of (*FPR*, *TPR*), which represent the average performance of bagging or single learners at nine different levels of class distribution.

In Fig. 3, the experimental results indicate that bagging MLP is superior to a single learner MLP on two out of 12 imbalanced data-sets, bupa and diabetes. Bagging MLP has a greater area than a single learner MLP, and therefore bagging MLP has better average performance than the single learner MLP on these two data-sets. The experimental results are consistent with the Wilcoxon Signed Rank Test on a *TPR* in Table 6.

Even though the average performance of bagging MLP is not superior to a single learner MLP, it has similar results to the single learner MLP on 10 imbalanced data-sets. Especially, on WDBC data-set, both bagging MLP and the single learner MLP perform extremely well on nine different imbalanced levels, as all nine pairs of (*FPR*, *TPR*) are close to the “ROC Heaven”, the upper left point (0, 1), and present almost 100% true positive and zero false positive.

6.3 Compare Bagging Predictors against One Another

This subsection compares bagging predictors against one another based on the Friedman with Post-hoc Nemenyi tests for comparison of all bagging predictors with two evaluation metrics, *G-mean* and *TPR*.

Table 7. Mean rank of Friedman Test for *G-mean* metric of Bagging Predictors

Mean Rank of Friedman Test for <i>G-mean</i> metric of Bagging Predictors						
Predictors	PART	MLP	NBTree	J48	RandTree	SVM
Mean Ranks	3.83	3.92	4.67	5.58	5.67	6.33
Predictors	NB	KNN	RepTree	Dtable	Dstump	OneR
Mean Ranks	6.50	6.83	7.08	8.75	9	9.83

Table 7 shows the average ranks of an evaluation metric *G-mean* for bagging predictors based on the corresponding base learners. The third and the last rows indicate the mean ranks of Friedman test results. The Friedman test is used for comparison of multiple learners. Firstly, we perform the Friedman test to compare the *G-mean* metric of 12 bagging predictors over 12 imbalanced data-sets and to obtain the mean ranks. As the Null Hypothesis is rejected, the Friedman test indicates there is at least a difference between the mean ranks of bagging predictors. Therefore the corresponding post-hoc Nemenyi test is required for additional exploration of the differences between the mean ranks to provide specific information on which mean ranks are significantly different from one another.

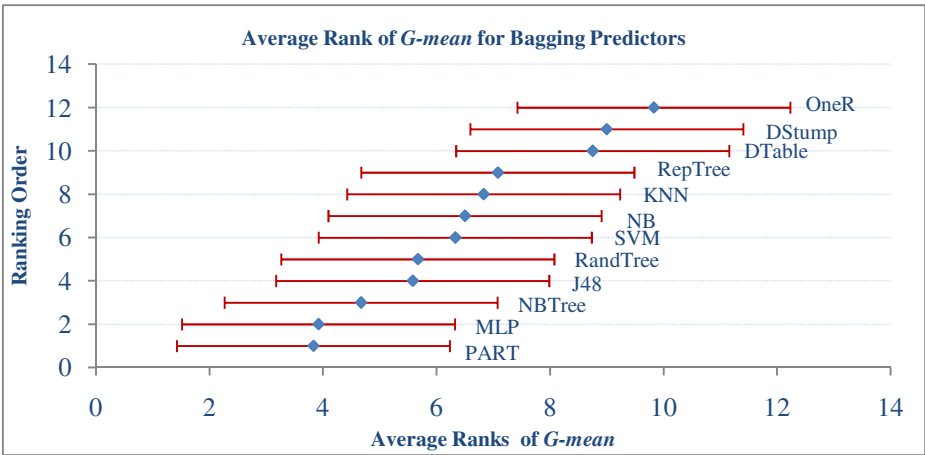


Fig. 4. Comparison of all Bagging predictors from Friedman and Post-hoc Nemenyi test, where *x*-axes indicate the mean rank of *G-mean* for bagging, the *y*-axes indicate the ascending ranking order of the Bagging predictors and the horizontal error bars indicate the “critical difference”

Fig. 4 reports the results of the Friedman with Post-hoc Nemenyi tests for comparison of the performance of all bagging predictors’ average ranks of *G-mean* on 12 imbalanced data-sets. The group of most robust base learners, PART and MLP contribute to the best bagging predictors; whereas the group of weakest learners, OneR and DStump lead to the worst bagging predictors. The performance of two bagging predictors is significantly different when the horizontal bars are not overlapping. There is a statistically significant difference between the two groups. As a result, one can conclude that the robustness of the base learners is an important factor for building accurate bagging predictors.

Table 8. Mean rank of Friedman Test for *TPR* Metric of Bagging Predictors

Mean Rank of Friedman Test for <i>TPR</i> Metric of Bagging Predictors						
Predictors	NBTree	MLP	PART	SVM	RdTree	RepTree
Mean Ranks	3.92	4.92	5.92	6.08	6.23	6.54
Predictors	Dtable	J48	KNN	NB	OneR	Dstump
Mean Ranks	6.65	6.65	7.54	7.62	7.65	8.27

Table 8 shows the average ranks of the Friedman Test for an evaluation metric *TPR* of bagging predictors. As the Null Hypothesis is accepted, the Friedman test indicates there is no difference between the mean ranks of bagging predictors. Therefore, the corresponding post-hoc Nemenyi test is not required for additional exploration of the differences between mean ranks to provide specific information on which mean ranks are significantly different from one another.

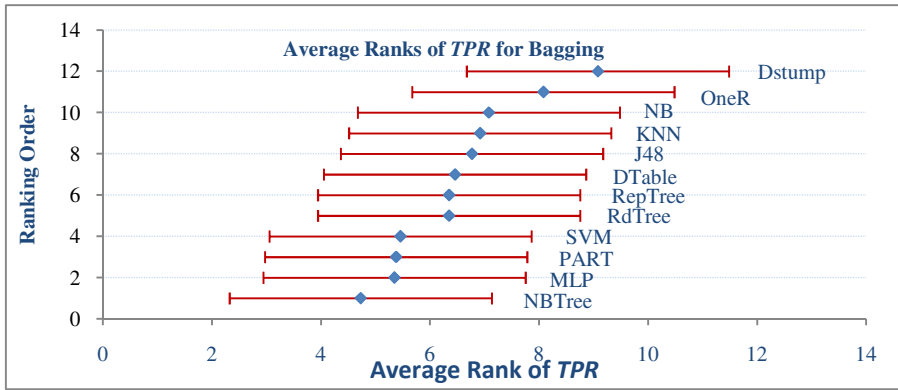


Fig. 5. Comparison of the *TPR* performance of all Bagging predictors, where *x*-axes indicate the mean rank of *TPR* for bagging predictors, and the *y*-axes indicate the ascending ranking order of the Bagging predictors

Fig.5 presents the mean ranks of *TPR* for all bagging predictors and demonstrates that NBTree, MLP, and PART are the learning algorithms with the best bagging performance on imbalanced data-sets, while DStump and OneR are the learning algorithms with the worst bagging performance on imbalanced data-sets. However, the mean ranks of *TPR* for bagging predictors are no significantly different from one another.

7 Conclusion

We introduced a new two-dimensional robustness and stability decomposition to provide a clear picture of the categorization of the base learners. This is significant as the predictive performance of bagging is influenced by the different types of base learners. We demonstrated that bagging is influenced by the combination of robustness and instability, pointing out that robustness is important for bagging to achieve a highly accurate prediction model. Our observations support our claims: the strongest base learners PART, MLP and NBTree can be used to build the best bagging predictive models, whereas the weakest learners, OneR and DStrump result in the worst bagging predictive models in the context of imbalanced learning. Our studies demonstrated that 4 out of 12 bagging predictors are statistically superior to single learners, including tree family learners, J48, RepTree and randTree, and a rule learner PART; this finding is based on both *G-mean* and *TPR* evaluation metrics over 12 imbalanced data-sets.

References

1. Breiman, L.: Bagging predictors. *Machine Learning* 24, 123–140 (1996)
2. Chan, P., Stolfo, S.: Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 164–168 (1998)

3. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. *Machine Learning* 30, 195–215 (1998)
4. Weiss, G.M., Provost, F.: Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* 19, 315–354 (2003)
5. Liang, G., Zhu, X., Zhang, C.: An Empirical Study of Bagging Predictors for Imbalanced Data with Different Levels of Class Distribution. In: Wang, D., Reynolds, M. (eds.) *AI 2011. LNCS (LNAI)*, vol. 7106, pp. 213–222. Springer, Heidelberg (2011)
6. Liang, G., Zhu, X., Zhang, C.: An empirical study of bagging predictors for different learning algorithms. In: *25th AAAI Conference on Artificial Intelligence, AAAI 2011*, pp. 1802–1803. AAAI Press, San Francisco (2011)
7. Quinlan, J.: Bagging, boosting, and C4. 5. In: *Proceedings of the National Conference on Artificial Intelligence*, pp. 725–730 (1996)
8. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research* 11, 169–198 (1999)
9. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning* 40, 139–157 (2000)
10. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36, 105–139 (1999)
11. Chawla, N.V.: Data mining for imbalanced datasets: An overview. In: *Data Mining and Knowledge Discovery Handbook*, pp. 875–886 (2010)
12. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance with nonuniform class and cost distributions. In: *Proceedings of AAAI 1997 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 57–63 (1997)
13. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453. Morgan Kaufmann (1998)
14. Ling, C.X., Huang, J., Zhang, H.: AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In: Xiang, Y., Chaib-draa, B. (eds.) *Canadian AI 2003. LNCS (LNAI)*, vol. 2671, pp. 329–341. Springer, Heidelberg (2003)
15. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7, 1–30 (2006)
16. Maloof, M.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: *ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, Washington, DC (2003)
17. Fawcett, T.: An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874 (2006)
18. Ng, W., Dash, M.: An Evaluation of Progressive Sampling for Imbalanced Data Sets. In: *Sixth IEEE International Conference on Data Mining Workshops, ICDM Workshops 2006*, pp. 657–661 (2006)
19. Provost, F., Fawcett, T.: Robust classification for imprecise environments. *Machine Learning* 42, 203–231 (2001)
20. Zeng-Chang, Q.: ROC analysis for predictions made by probabilistic classifiers. In: *Proceedings of ICMLC 2005*, pp. 3119–3124 (2005)
21. Witten, I.H., Frank, E.: *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco (2005)
22. Merz, C., Murphy, P.: *UCI Repository of Machine Learning Databases* (2006)

Exploiting Concept Clumping for Efficient Incremental News Article Categorization

Alfred Krzywicki and Wayne Wobcke

School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052, Australia
{alfredk,wobcke}@cse.unsw.edu.au

Abstract. In this paper, we introduce efficient methods for incremental multi-label categorization of documents. We use concept clumping to efficiently categorize news articles into a hierarchical structure of categories. Concept clumping is a phenomenon of local coherences occurring in the data and it has been previously used for fast, incremental e-mail classification. We extend the definition of clumping and introduce additional clumping metrics specifically for multi-label document categorization. We present three methods for incremental multi-label categorization that exploit concept clumping and make use of thresholding techniques and a new term-category weight boosting method. Our methods are tested using the Reuters (RCV1) news corpus and the accuracy obtained is comparable to some well known machine learning methods trained in batch mode, but with much lower computation time.

Keywords: multi-label document categorization, text mining, concept drift.

1 Introduction

Fast, incremental document categorization is important in applications assisting individual users with organizing and labelling text documents such as news articles, e-mails, and paper and book abstracts. E-mail messages are typically classified into one category, commonly called a folder. Many document types, however, for example news articles, need to be classified into more than one category. In this case categories become document labels, rather than storage folders. The labels may form a hierarchical structure with more general labels as parents and the most specific ones as leaves of the structure. This is the case for the Reuters news article set (referred to as RCV1), being used for evaluation in this paper.

While there are many research reports on multi-label document categorization in batch mode (e.g. [13,8]), we are not aware of any work on efficient multi-label document categorization in true incremental mode. In Krzywicki and Wobcke [6,5], we introduced three methods for fast, incremental, single label document categorization: Simple Term Statistics (STS), Local Term Boosting (LTB) and Weighted STS (WSTS). Both LTB and WSTS adjust their term-category weights to follow local trends in the data and are able to learn the weight boosting factor dynamically. In this paper, we extend these methods and apply them to classify each document into one *or more*

categories, i.e. multi-label document categorization. The main challenge to address here is the determination of the categories and their number for each document¹. The most common technique employed for this task is thresholding, where categories are weighted according to some algorithm, and the document is labelled with categories whose weights exceed the threshold θ .

In this paper, we adapt our methods for single label classification to the incremental multi-label setting, by introducing and evaluating a number of thresholding techniques where the threshold value is learned dynamically. Another important contribution of this paper is the method of learning the term-category weight boosting factor together with the threshold, as these two variables are interdependent.

All clumping definitions given in Krzywicki and Wobcke [6] are applicable for multi-label document categorization. In this paper, however, in Section 3, we introduce additional clumping measures used for accuracy analysis of multi-label categorization and show its dependency on term-category clumping in the RCV1 dataset. Our methods are not directly comparable with existing published methods, since as far as we know, this is the first paper to address experimental evaluation on the RCV1 dataset in strictly incremental mode. Nevertheless, the accuracy of our incremental methods is similar to those obtained by other researchers using batch mode, but with the advantage of a much faster runtime.

The rest of the paper is structured as follows. In the next section, we describe related work, then in Section 3 we cover concept clumping and related metrics, specific to multi-label document categorization. Section 4 describes our document categorization methods and related algorithms. In particular, Section 4.2 covers thresholding algorithms and Section 4.3 discusses term-category weight boosting with the threshold as a parameter used in the boosting algorithm. In Section 5 we describe the experimental evaluation of STS, LTB and WSTS using the new thresholding methods. Finally, Section 6 summarizes the paper.

2 Related Work

A variety of methods have been researched for multi-label document categorization. Yang [11] reports a study of text classification algorithms on the Reuters-21450 dataset, which includes a number of unlabelled documents, at least some of which should have labels. In this evaluation, k-Nearest Neighbour (k-NN) was shown to perform well. The decision of whether a document needs to be classified or not was made by setting a threshold on an array of values representing learned weights for each document-category pair. A document is classified by the system if the weight is above the threshold, otherwise it is not classified by the system. An optimal threshold is calculated on the training set and used on test documents in a typical batch fashion. As shown in Krzywicki and Wobcke [6,5] our methods can also use the threshold strategy for separating the messages into classified and unclassified, but the threshold is adjusted dynamically in the process of incremental classification, allowing for optimal coverage of classified messages.

¹ In this paper, the terms “category” and “label” are used interchangeably.

A range of now widely used thresholding techniques have been reviewed by Yang [12]. RCUt selects the top t documents ranked in some way, where t is an integer which can be fixed or tuned automatically in the process of training, for example to optimize a global performance measure. PCut assigns the “YES” decision to a number n of top-ranked documents for each category, rather than a number of categories to a document. The number n depends on the probability of the given category and an additional parameter, which can be tuned in a similar way to RCUt. The third method, SCut, is based on learning an optimal threshold for each category.

A slightly different technique, RinSCut is proposed by Lee *et al.* [7]. Documents are ranked by a similarity score and two thresholds are used: ts_{top} and ts_{bottom} . If a document scores above ts_{top} or below ts_{bottom} , the decision is “YES” or “NO” respectively. For documents that fit in between the thresholds, the decision is based on a learned, category specific threshold in the same way as SCut.

More recent results on the use of the RCV1 dataset with a Perceptron based method (which is also related to our methods) are presented by Gkanogiannis and Kalamoukis [3]. When trained and tested in batch mode, the method shows good accuracy, comparable with SVM, and relatively good time performance (expressed as the number of iterations). Because of the batch mode used by the authors, their results are not directly comparable with ours.

3 Concept Clumping for Multi-label Categorization

In Krzywicky and Wobcke [6] we defined category and term-category clumping for single category classification. According to these definitions, a category clump for category f is a contiguous sequence of documents (excluding the first one) classified in f . A term-category clump for term t and category f is an f category clump in the subsequence of all documents that contain t . In this paper, we extend the definition of term-category clumping, making it more suitable for multi-label document categorization.

Definition 1. Given a term t , a set of categories F^t such that $|F^t| = k$, and the sequence D^t of all documents containing t each categorized into at least one of the categories from F^t , a **k -term-category clump** $D^{t,f} = \{d_1^{t,f}, \dots\}$ for term t and categories in F^t is a maximal contiguous subsequence of D^t such that $d_0^{t,f} \in D^t$ where $d_0^{t,f}$ is the document preceding $d_1^{t,f}$ in D^t , and each $d_i^{t,f} \in D^t$.

The k -term-category clumps span over instances in a fixed set of categories containing a given term. To give a simple example, let us assume we have three instances: $i1=(t1,t2,t3)$ in category $f1$, $i2=(t1,t4,t5)$ in category $f2$, and $i3=(t1,t6,t7)$ in category $f3$. The instances $i1$ and $i2$ form a $k = 2$ clump for term $t1$ (assuming some instance $i0$ is also in the same sequence), because $t1$ is in two categories for these two instances. However, since $i3$ is in $f3$, the $t1$ term must indicate $f3$. Since for $k = 2$, $t1$ can only indicate two categories for the same clump, $i3$ breaks the existing clump (but may start another one with subsequent instances).

We also provide a definition of the clumping ratio consistent with the definitions given above.

Definition 2. The *theoretical maximal number of clumps* $c_{max}(d)$ for an instance d in an instance sequence D is defined as $c_{max}(d) = |F^d|$ for category clumping, and $c_{max}(d) = |T^d| * |F^d|$ for term-category clumping, where F^d is the set of categories to which d is assigned, and T^d is a set of terms in d .

Definition 3. The *instance clumping ratio* m is defined as $m = c(d)/c_{max}(d)$, where $c(d)$ is the actual number of clumps containing the instance d , and $c_{max}(d)$ is the theoretical maximal number of clumps for instance d .

4 Categorization Methods

In this section, we briefly describe the base algorithms Simple Term Statistics (STS), Local Term Boosting (LTB) and Weighted Simple Term Statistics (WSTS), Krzywicki and Wobcke [5][6], then define new thresholding techniques and term-category weight boosting methods designed to work with multi-label document categorization.

4.1 Base Algorithms

The STS method maintains an array of weights, one for each term and category for all documents in dataset. Each weight is calculated as a product of two numbers: a term ratio (a “distinctiveness” of term t over the document set, independent of category-specific measures), and the term distribution (an “importance” of term t for a particular category). Each term ratio formula is given a letter symbol from a to d and term distribution formulas are numbered from 0 to 8. For example, $M_{b2} = \frac{1}{N_{ft}} * \frac{N_{dtf}}{N_{dt}}$, where N_{ft} is the number of categories where term t occurs, N_{dtf} is the number of documents containing term t in category f and N_{dt} is the number of documents in training set containing term t . This method performs well across most of the datasets used in this research, therefore it is used in the experimental evaluation. The predicted category f_p for document d is defined as follows:

$$f_p = \operatorname{argmax}_f(w_f) \text{ when } \max_f(w_f) \geq \theta \quad (1)$$

where $w_f = \sum_{t \in d} w_{t,f}$ and θ is a threshold.

Similar to STS, LTB maintains an array of weights $w_{t,f} (|T| \times |F|)$ for each term t and each category f . The difference is in the way the term weights for each category $w_{t,f}$ are calculated. Initially, all weights are initialized to a constant value of 1.0. After processing the current document d , if the predicted category f_p is incorrect, weights of term in the target category are increased, while weights in the predicted category are decreased. The amount of weight increase/decrease depends on boosting factor b used to represent the speed of weight adjustment. Weights are calculated for both the predicted category f_p and the target category f_t using the formula $w_f = \sum_{t \in d} w_{t,f}$.

The WSTS method is a combination of STS and LTB. It maintains two types of term-category weights: one for STS, which we will denote by $\rho_{t,f}$, and one for LTB, denoted as before by $w_{t,f}$. Again the predicted category f_p for document d is calculated in the same way as for STS, except that the total term-category weight is obtained by multiplying STS weights by LTB weights, i.e. $w_f = \sum_{t \in d} (\rho_{t,f} * w_{t,f})$. As in LTB, the boosting factor b is used to adjust the weights $w_{t,f}$ after the prediction is made.

4.2 Thresholding Techniques

Three thresholding techniques have been evaluated. The first (*Th1*) calculates threshold values optimized to maximize the F1 measure using the estimation-maximization approach. The other two thresholding algorithms attempt to find an optimal number of labels by averaging over all past instances (*Th2*) and in a sliding window (*Th3*). All these methods are effectively variants of *RCut* [12]. *RCut* assigns a number t of top rated categories to a document, where t is a parameter that can be tuned to optimize a performance measure, usually F1. This parameter can change its value independently from one instance to another. For this reason, this technique can be used in incremental processing (which is pointed out in Yang [12]). Other thresholding methods described in that paper (*SCut* and *PCut*) tune parameters on a training set of documents specific to batch learning, therefore are not tested here.

F1 Optimized Thresholding Method (*Th1*)

The *Th1* method is different from *RCut* in that the t parameter is not the number of categories to assign “YES” to, but the score itself. All Categories with the score (the total weight for a category in case of STS, LTB and WSTS) equal to or exceeding the parameter value are assigned to the document being categorized. The novelty of the method is also an incremental learning of the threshold to maximize the F1 measure, given as $F1 = 2 * Recall * Precision / (Recall + Precision)$. Let c be the number of correctly predicted categories, l the number of all predicted categories and t the number of target categories in a single document. Since $Recall = c/t$ and $Precision = c/l$, at the document level we have

$$F1 = \frac{2 * c}{l + t}. \quad (2)$$

In our calculations, however, we often express F1 calculated over a set of documents as a microaverage measure in the following form:

$$F1 = \frac{2 * \sum_1^n c}{\sum_1^n l + \sum_1^n t} \quad (3)$$

where the summation is over n documents in the dataset.

The threshold optimization method is described by the following algorithm.

Algorithm 1 (F1 Optimized Threshold Algorithm)

- 1 Categorize previous instance
- 2 $w_{fmax} := \max_f(w_f)$
- 3 $w_{fmin} := \min_f(w_f)$
- 4 $F_{cur} := \{\}$
- 5 $F1_{max} := 0$
- 6 $th_{best} := 0$
- 7 Sort categories F in descending order of w_f
- 8 **forall** $f \in F$
- 9 $F_{cur} := F_{cur} \cup f$
- 10 $n_f^l := n^l + |F_{cur}|$
- 11 $n_f^c := n^c + |F_{cur} \cap E_t|$

```

12   $n_f^t := n^t + |F_t|$ 
13   $F1 := 2 * n_f^c / (n_f^l + n_f^t)$ 
14  if  $F1 > F1_{max}$ 
15     $F1_{max} := F1$ 
16    if  $w_{fmax} = w_{fmin}$ 
17       $th_{best} := 0$ 
18    else
19       $th_{best} := (w_f - w_{fmin}) / (w_{fmax} - w_{fmin})$ 
20    endif
21  endif
22 endfor
23  $threshold := (threshold * NumDocs + th_{best}) / (NumDocs + 1)$ 

```

In the above algorithm, w_f is the score calculated for category f during document categorization, F is the set of all categories, $n^l = \sum_1^n l$ is the sum of numbers of categories in each document as categorized by the learner up to and including the current instance, $n^c = \sum_1^n c$ is the sum of numbers of categories in each document as correctly categorized by the learner up to and including the current instance, $n^t = \sum_1^n t$ is the sum of numbers of actual categories in each document up to and including the current instance, and F_t is the set categories (labels) into which the current document should be classified (target categories).

In the estimation stage (line 1), a document is categorized (estimated) based on the previously calculated threshold. The rest of the algorithm is the maximization stage, where a new value of the threshold is calculated to maximize the F1 measure for the recently categorized instance. Firstly, minimum and maximum scores for all categories are found and the set of categories F is sorted in descending order of the score. In the next step, a set of categories F_{cur} is incrementally constructed by adding categories from F . After each addition of a category from F to F_{cur} , F1 is calculated and compared with its highest value obtained so far. If the current F1 exceeds $F1_{max}$, the new $F1_{max}$ and the normalized weight of that category are stored in memory. This weight becomes progressively smaller until the maximum value of F1 is reached, and becomes an estimation of the new threshold value. The fact that the algorithm does not exit the loop at this point, but continues until all categories are tried, helps in overcoming possible local maxima of F1. In the last step (line 23), an effective threshold is calculated as a running average for all instances categorized so far.

Category Rank Thresholds (*Th2* and *Th3*)

The *Th2* and *Th3* thresholding methods are similar to *RCut* [12] in that the threshold is a rank above or equal to which all categories are assigned to an instance. The difference is in the way these thresholds are calculated. Firstly, the thresholds are adjusted incrementally to approximate an average number of categories assigned to documents. Secondly, these thresholds are converted to real numbers and used by term-category weight adjustment. Effectively *Th2* and *Th3* maintain two types of thresholds: the category rank threshold and the category weight threshold.

Determining the threshold by taking into account numbers of categories in each document is based on the assumption that this number is not completely random, but follows some clumping patterns that occur when documents are presented in time order. Examination and analysis of documents confirmed the correctness of this assumption.

Firstly, we determine the number-of-categories clumping ratio, which follows the category clumping ratio defined in Section 3, except that instead of sequences of documents with categories of the same name we track sequences of documents with the same number of categories (labels) assigned to each document. In the first 23149 documents in the RCV1 dataset used in this paper, the number-of-categories clumping ratio is 0.396. 47% of documents have 3 categories and 26% have 2 categories. For example, in a range of 150 RCV1 documents starting at document 2614, containing one large clump of instances with 2 categories and a few small clumps with 3 categories. A closer look at the news messages corresponding to these documents revealed that the categories in the 2-category clump are all related to a sport event and the 3-category clumps are grouped around corporate financial topics. The above observation is the basis of the following algorithm, which has two variants, for the calculation of *Th2* and *Th3* thresholds.

Algorithm 2 (Threshold Algorithm Based on Number of Categories)

```

1  Update term-category statistics (STS and WSTS)
2  Calculate category scores  $w_f$ 
3   $w_{fmax} := \max_f(w_f)$ 
4   $w_{fmin} := \min_f(w_f)$ 
5   $F_p := \{\}$ 
6  Sort categories  $F$  in descending order of  $w_f$ 
7   $i := 0$ 
8  do until  $i < AVGNcut$ 
9     $F_p := F_p \cup f_i$ 
10    $i := i + 1$ 
11 enddo
12  $f := f_{AVGNcut}$ 
13 if  $w_{fmax} = w_{fmin}$ 
14    $threshold := 0$ 
15 else
16    $threshold := (w_f - w_{fmin}) / (w_{fmax} - w_{fmin})$ 
17 endif
18 Adjust term-category weights (LTB and WSTS)
19 {Th2}
20  $AVGNcut := (AVGNcut * NumDocs + |F_t|) / (NumDocs + 1)$ 
21 {Th3}
22  $putFIFO(s_t, |F_t|)$ 
23  $AVGNcut := Average(s_t, n)$ 
24  $AVGNcut := round(AVGNcut)$ 

```

In the above algorithm, F is a set of categories, F_p is the set of categories predicted for the current instance, F_t is the set of actual (target) categories for the current

instance, $AVGNcut$ is an average number of categories calculated over all past instances (method *Th2*) or a window of the last n instances (method *Th3*), w_f is the weight of the category f , $f_{AVGNcut}$ is the category with rank $AVGNcut$, $NumDocs$ the running number of documents, and s_t is a FIFO (First In First Out) memory of number of categories in the last n instances.

Unlike method *Th1*, this algorithm does not use the weight threshold but assigns an average number of top rated categories in the prediction phase, where the average is calculated over all past examples (method *Th2*) or just a window of n past examples (method *Th3*). The value of n is chosen to be small (4 in our experiments, for larger values *Th2* becomes close to *Th3*). Also notice that the thresholds in this algorithm are not explicitly optimized with respect to the F1 measure, instead they follow the target number of top ranked categories.

The above algorithm shows the whole cycle of category prediction for a new instance. It starts with updating the term statistics for STS and WSTS and then proceeds to assigning top weighted $AVGNcut$ categories to the document in lines 6–11 using category rank threshold.

In the next stage, a new weight threshold is calculated in lines 12–17 as a weight of the category, which has the cut-off rank $AVGNcut$. This weight threshold is required by the term-category weight adjustment algorithm (line 18), described in detail in the next section. Finally, a new category rank threshold is calculated in lines 19–24 and rounded to the nearest integer. As in the *Th1* algorithm, the weights to calculate the threshold are normalized by their minimal and maximal values.

4.3 Term-Category Weight Boosting

In Krzywicki and Wobcke [6], an algorithm for the boosting of term-category weights was introduced. In that algorithm, the boosting factor b is updated dynamically to approximate a value which would increase the weight of the target category to make it the top category. The threshold was compared to this weight to determine whether a document should be classified or not. In multi-label classification, the boosting factor should be calculated to approximate a value that would increase the target category (label) weight above the threshold. For this reason, the threshold calculation is closely related to the update of the boosting factor.

We derive the formula for the boosting factor adjustment δ_b by comparing the adjusted category weight with the threshold: $w_f + \delta_b = threshold + \epsilon$ where ϵ is a small value chosen to make the adjusted weight slightly greater than the threshold. Since the threshold is maintained as a normalized value with regards to the category weights, the left hand side of the equation needs also to be normalized: $(w_f + \delta_b - w_{fmin}) / (w_{fmax} - w_{fmin}) = threshold + \epsilon$. Therefore

$$\delta_b = (threshold + \epsilon) * (w_{fmax} - w_{fmin}) + w_{fmin} - w_f. \quad (4)$$

The weight boosting algorithm is presented below.

Algorithm 3 (Adjusting Term-Category Weights)

```

1  forall categories  $f$ 
2     $\delta_b = (\text{threshold} + \epsilon) * (w_{f_{max}} - w_{f_{min}}) + w_{f_{min}} - w_f$ 
3     $b = (b * N + \delta_b) / (N + 1)$ , where  $N$  is current number of instances
4  endfor
5  forall categories  $f$ 
6    forall terms  $t$  in document  $d$ 
7      if  $f \in F_t$  and  $f \notin F_p$  then
8         $w_{t,f} := w_{t,f} * (1 + b / (w_f + b))$ 
9      endif
10     if  $f \in F_p$  and  $f \notin F_t$  then
11        $w_{t,f} := w_{t,f} * (1 - b / (w_f + b))$ 
12     endif
13   endfor
14 endfor

```

where F_t is the set of target categories for document d , F_p is the set of predicted categories for document d , w_f is the weight of category f , and $w_{t,f}$ is the weight associated with term t and category f .

The difference from the algorithms for single label categorization given in [6] is that the boosting factor b is adjusted as an average over adjustments for each category. The boosting factor adjustment δ_b is calculated with respect to the threshold rather than the maximal category weight. Term-category weight boosting is also done for all categories to find whether category f is in the target or predicted categories for d .

5 Experimental Evaluation

In this section we evaluate the extended STS, LTB and WSTS methods with the different thresholding techniques on the RCV1 document set. The task here is to assign a number of categories to each document to match the number and names of target categories.

5.1 The RCV1 Datasets

For evaluation of the multi-label categorization methods, we use the first 23149 of the RCV1 document set [8]. All documents are classified into 101 categories, with one or more categories per document. Because of the hierarchical structure of categories, all parent categories of a category assigned to a document are also assigned to that document.

The document set consists of three types of files which are read and processed separately. The category (topic) hierarchy file contains all category names and their parents. The document-category file contains a mapping from the document number to its categories. The document contents file contains document number and a list of terms occurring in the document. All terms are stemmed and with stop words removed.

Table 1 lists basic statistics and k -term-category clumping ratios for the RCV1 dataset, considering *All Categories* and *Leaf Categories Only*, for various values of k . The equal

Table 1. Statistics of the RCV1 Dataset

	RCV1 All	RCV1 Leaf
#categories	101	101
#documents per category	729	329
#categories per document	3.18	1.44
% of documents in largest category	47	8.4
avg 1-term-cat. clumping ratio	0.009	0.104
avg 2-term-cat. clumping ratio	0.098	0.189
avg 3-term-cat. clumping ratio	0.233	0.254
avg 4-term-cat. clumping ratio	0.288	0.305

number of categories in both variants indicates that some documents are labelled with the top categories. For example, the top category GCAT is a leaf label for 735 documents and ECAT for only 15 documents.

Each document is categorized into about 3 categories on average for the full hierarchy, and about 1.44 categories for the leaf only categorization, for which parent labels are not automatically included (for *RCV1 All*, if a document is categorized into f , it is also categorized into all parents of f). For this reason, there are significant differences in clumping between these two cases. As can be expected, the term-category clumping ratios are much higher for up to $k = 3$. After that point the term-category clumping ratios become comparable with RCV1 All Categories. The reason for this difference is that with parent categories always included it is harder to find terms that can only occur in 1 or 2 categories.

When comparing the term-category clumping ratios for different values of k , the RCV1 All Categories values show sharp increase until $k = 3$. For RCV1 Leaf Only the corresponding k is 2. These k values are roughly equal to the average number of categories per instance for each case. The term-category ratios for the RCV1 dataset are the lower than those on the Enron dataset reported in [6]. Note that a k -term-category clump is necessarily contained in a $k+1$ -term-category clump, hence the clumping ratios increase as k increases.

5.2 Experimental Setup

The results discussed in this paper are expressed mainly as an F1 measure at the document level (Equation 2) or as microaverage over a set of documents (Equation 3).

There are two reasons for using the F1 measure: (i) due to the variable number of labels for each document, it is important to measure both precision and recall that are interdependent and combined in the F1 measure, and (ii) the F1 measure has been commonly used in the literature for presentation of results on the RCV1 dataset ([18,10]). We prefer to use microaveraging as it is more suitable for incremental classification, since it updates counts incrementally after each instance. In contrast, macroaveraging uses separate counts for each category, which are not known in advance. Also, the distribution of instances into categories changes dramatically as the incremental categorization task progresses (e.g. there is only one instance in a newly occurring category),

therefore the macroaveraged F1 measure is less reliable in reflecting the current classification accuracy.

Testing has been done using incremental training on previous document and testing on the next document for all three threshold algorithms detailed in Section 4.2 and for both variants of the dataset: with All Categories and with Leaf Categories Only.

5.3 Discussion of Results

Table 2 shows the results of microaveraged F1 for the LTB, STS and WSTS methods for both All Categories and Leaf Categories Only. As expected, F1 is much higher in the All Categories case due to the presence of parent categories in each document. Since there are only 4 top categories, and all parent categories are automatically included, with 47% of documents in at least one of the top categories, the prediction is much easier for this variant. In this sense, the Leaf Categories Only case seems to be a more genuine prediction task. In the literature, however, the All Categories variant is used more often than Leaf Categories Only.

Table 2. Microaveraged F1 Results on RCV1 Document Set

Method	All Categories			Leaf Categories Only		
	<i>Th1</i>	<i>Th2</i>	<i>Th3</i>	<i>Th1</i>	<i>Th2</i>	<i>Th3</i>
LTB	0.5623	0.6493	0.6459	0.5334	0.5185	0.5302
STS	0.574	0.5902	0.5883	0.4946	0.482	0.4815
WSTS	0.6768	0.6627	0.6532	0.565	0.5094	0.5114

The best F1 of 0.6768 is achieved using the WSTS method for the F1 optimized threshold algorithm (*Th1*). The other two thresholding methods, based on the clumping of number of categories for each instance, also give good results, only a couple of percent lower than the top method. This is surprising, since these methods are not explicitly optimized for the F1 measure. The presented results are lower than the best results obtained by other researchers on the RCV1 document set. The microaveraged F1 measure obtained by Lewis *et al.* [8] is 0.816 which is the best published result to our knowledge. This result, however, was obtained in batch mode by the best classifier (SVM), where a number of classifiers were trained on the 23149 documents before any testing was done on other datasets. During the training phase, supporting algorithms, such as feature selection and the thresholding technique, were carefully selected and tuned for each classifier. This approach is not feasible in our case, where we process the documents in strictly incremental fashion rather than using batch processing. The second reason is that our methods rely on clumping ratios that for this dataset may not be sufficient to obtain better results. Our methods, however, as shown later in this section, can achieve much higher F1 on sequences of documents where this clumping is higher.

We compare the computation time of our methods with results obtained by Granitzer [4] which is the only published work known to us containing training times for the RCV1 dataset with a similar number of training instances to our training set, but using batch mode with multiple iterations. Out of a number of methods used (Centroid Booster [4], AdaBoost [2] and Rocchio [9]), the Rocchio algorithm is fastest with

results given on the Leaf Categories Only variant of the RCV1 dataset (times for All Categories are not given). Our WSTS method is about 2 to 3 times faster than Rocchio with comparable F1 measure (WSTS 0.565, Rocchio 0.538), taking into account differences in dataset sizes, CPU power and memory size.

Table 3 shows correlations, for both category measures, between the F1 measure and some of the clumping metrics over all instances of 23149 documents. The correlation is quite substantial, reaching 50% or more for most methods and clumping metrics. The k -term-category clumping correlations tend to be higher for higher values of k , which may be an indication of the frequent existence of terms common to more than one category.

Table 3. Correlations Between F1 and Clumping Ratios

Method	LTB			STS			WSTS		
	<i>Th1</i>	<i>Th2</i>	<i>Th3</i>	<i>Th1</i>	<i>Th2</i>	<i>Th3</i>	<i>Th1</i>	<i>Th2</i>	<i>Th3</i>
All Categories									
Term-cat. clump ratio									
k=1	0.209	-0.099	0.008	0.235	-0.036	0.102	0.067	0.206	0.024
k=2	0.519	0.111	0.401	0.551	0.290	0.530	0.526	0.403	0.428
k=3	0.749	0.631	0.769	0.809	0.724	0.773	0.820	0.738	0.770
k=4	0.735	0.628	0.759	0.778	0.707	0.711	0.795	0.735	0.752
Leaf Categories Only									
Term-cat. clump ratio									
k=1	0.676	0.703	0.710	0.682	0.704	0.704	0.644	0.679	0.681
k=2	0.689	0.742	0.750	0.687	0.723	0.720	0.675	0.696	0.697
k=3	0.687	0.743	0.751	0.689	0.723	0.719	0.678	0.698	0.699
k=4	0.686	0.743	0.751	0.690	0.721	0.718	0.679	0.698	0.700

Another interesting observation can be made about k -term-category clumping. As can be seen in Table 1, the k -term-category clumping for the All Categories measure increases sharply until about 3 common categories ($k = 3$), then the increase is much slower. This corresponds to the correlation numbers (Table 3) for the All Categories measure, where the correlation peaks at about 3 common terms, whereas for the Leaf Categories Only measure, the correlation does not change much at about $k = 2$. This roughly corresponds to the average number of categories per document for these measures.

Figure 1 shows a sub-sequence of the documents with high term-category clumping. This sequence was selected manually and it corresponds to two days in August 1996, when the number of news articles related to sport events was very high. The first graph of the figure shows the result for the All Categories measure, and the next graph shows the result for the Leaf Categories Only measure. In this sequence, the term-category clumping is highly correlated with the F1 measure for all documents in the sequence, and the F1 measure reaches over 90%. We believe that this is a good indicator that

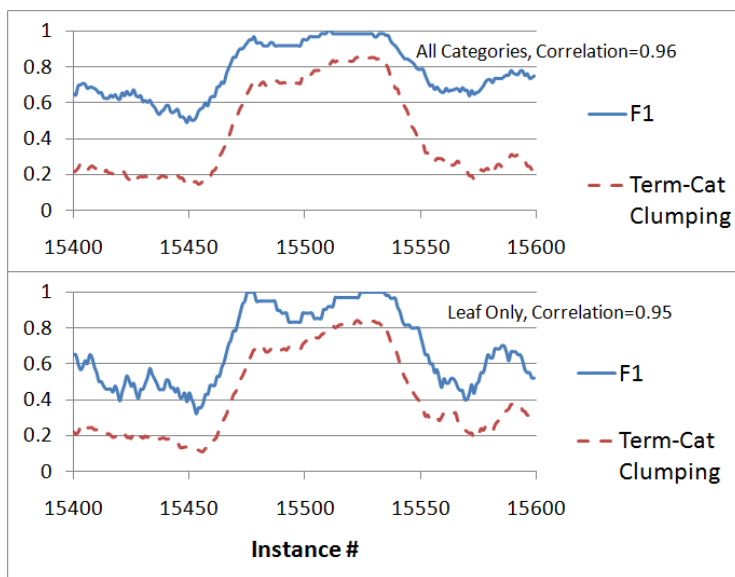


Fig. 1. F1 Measure and Term-Category Clumping Ratios

the LTB, STS and WSTS methods can achieve high precision and recall as long as the term-category clumping is also high.

6 Conclusion

In this paper, we addressed challenges related to document categorization into multiple categories that form a hierarchical structure. Our general methods, namely STS (Simple Term Statistics), LTB (Local Term Boosting) and WSTS (Weighted STS), introduced and tested in our previous research, are shown to work well in the multi-label setting using thresholding techniques and a method for learning the boosting factor for LTB and WSTS. We based our thresholding methods on a well known thresholding technique, called *RCut*, Yang [12]. The *Th1* algorithm uses category weight as a threshold and is optimized for the F1 measure. The other two thresholding algorithms, *Th2* and *Th3*, use clumping of the number of categories in sequences of documents. We showed that all three thresholding algorithms provide good results in terms of the F1 measure. *Th2* and *Th3*, while not specifically optimized for the F1 measure, nevertheless give results comparable with *Th1* on this measure.

We analysed the correlations between the precision/recall measure F1 and various clumping metrics taken over the entire dataset as well as on selected sequences of documents. The analysis showed that the instance based F1 measure is highly sensitive to the clumping of terms and categories occurring in the data and can reach very high values over 90% on sequences where the clumping is high. In addition, the experimental results also show that LTB, STS and WSTS, when applied to this document set, are able

to detect and utilize multiple, complex dependences between terms and categories and assign multiple categories to documents with high accuracy.

We compared the computational efficiency of our methods with runtimes for a similar dataset drawn from the same corpus but in the batch mode. Our incremental methods are much faster and more suitable for online applications. However, due to generally lower clumping in the RCV1 dataset, the recall/precision results are not as high as those obtained in the batch learning mode, where all parameters are carefully selected and tuned on a fixed training set. We believe that our methods compensate for this deficiency with increased learning speed.

Future work includes applying similar techniques to other domains and on other news datasets, and incorporating the methods into a practical news filtering application.

References

1. Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. *Information Retrieval* 11, 287–313 (2008)
2. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: *Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*, pp. 148–156 (1996)
3. Gkanogiannis, A., Kalamboukis, T.: A Perceptron-Like Linear Supervised Algorithm for Text Classification. In: Cao, L., Feng, Y., Zhong, J. (eds.) *ADMA 2010, Part I. LNCS*, vol. 6440, pp. 86–97. Springer, Heidelberg (2010)
4. Granitzer, M.: *Hierarchical Text Classification Using Methods from Machine Learning*. Master's thesis, Institute of Theoretical Computer Science (IGI), Graz University of Technology (2003)
5. Krzywicki, A., Wobcke, W.: Incremental E-Mail Classification and Rule Suggestion Using Simple Term Statistics. In: Nicholson, A., Li, X. (eds.) *AI 2009. LNCS*, vol. 5866, pp. 250–259. Springer, Heidelberg (2009)
6. Krzywicki, A., Wobcke, W.: Exploiting Concept Clumping for Efficient Incremental E-Mail Categorization. In: Cao, L., Zhong, J., Feng, Y. (eds.) *ADMA 2010, Part II. LNCS*, vol. 6441, pp. 244–258. Springer, Heidelberg (2010)
7. Lee, K.-H., Kay, J., Kang, B.-H., Rosebrock, U.: A Comparative Study on Statistical Machine Learning Algorithms and Thresholding Strategies for Automatic Text Categorization. In: Ishizuka, M., Sattar, A. (eds.) *PRICAI 2002. LNCS (LNAI)*, vol. 2417, pp. 444–453. Springer, Heidelberg (2002)
8. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5, 361–397 (2004)
9. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, Englewood Cliffs (1971)
10. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Learning hierarchical multi-category text classification models. In: *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pp. 744–751 (2005)
11. Yang, Y.: An evaluation of statistical approaches to text categorization. *Information Retrieval* 1, 69–90 (1999)
12. Yang, Y.: A study of thresholding strategies for text categorization. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 137–145 (2001)

Extracting Rocks from Mars Images with Data Fields

Shuliang Wang and Yashen Chen

International School of Software, Wuhan University, Luojia Hill, Wuhan 430079, China
slwang2005@whu.edu.cn

Abstract. In this paper, a novel method is proposed to extract rocks from Martian surface images by using data field. Data field is given to model the interaction between two pixels of a Mars image in the context of the characteristics of Mars images. First, foreground rocks are differed from background information by binarizing image on rough partitioning images. Second, foreground rocks is grouped into clusters by locating the centers and edges of clusters in data field via hierarchical grids. Third, the target rocks are discovered for the Mars Exploration Rover (MER) to keep healthy paths. The experiment with images taken by MER Spirit rover shows the proposed method is practical and potential.

1 Introduction

It is difficult to extract rocks from intensity images of planetary surface because there is no uniform morphology, color, texture or other quantitative measures to characterize them from other features [1][2][3]. With the development of space exploration technology, the requirements towards the capability of analyzing data from outer space exploring tools are also increasing, especially for such planets as Mars on which creatures might live. It is significant to analyze the imagery data from Mars exploration, especially differing rocks, e.g. rocks, gravels or creatures, from background information in an image. In future Mars rover missions, the rovers will travel much longer distances than those have achieved by the Mars Exploration Rover (MER) Spirit and Opportunity [1]. Autonomous foreground object extraction will be one of the major parts of an image analysis system for Mars exploration. On-board autonomous image analysis is highly desirable in future Mars rover missions [2]. Compared to the normal images, Mars images show its singular features, e.g. less color, monotonous scenery, unsteady light source, and so on. In addition, objects are often covered by dust, occluded each other, becoming blurred in the distance or partially embedded in terrain [1]. Rocks are one of the major features exposed on Martian surface, if rocks are automatically extracted from Mars images, people are able to separate the focus, and carry out further study on interested partitions.

It is valuable for hazard avoidance and rover localization in rover missions to automatically extract rocks from Mars images. Some rocks, e.g. rocks, are one of the major obstacles for rover traversing and even endanger the rovers' safety if they can't be detected correctly in advance. Rocks are also the ideal tie points for vision-based rover localization and navigation [3]. The automatic extraction of rocks may

benefit the finding available route automatically for Mars rover, and even on non-stone object (e.g. creatures) discovery. In addition, rocks may be used to judge the image information content for helping image compress and prioritize data transmission from Mars to Earth, or for understanding the planet's chemical composition, geologic environment, and climate mechanics [4].

Current Mars image processing is developing rapidly. The representative methods include edge detection, Fourier transformation, wavelet transformation, threshold extraction, and active contour recognition. In edge detection there are Sobel, Canny, and LoG. For solving the variable illumination problem, histogram equalization, homomorphic filter was given [5]. The modeling requires that the image has same features [6][7]. Histogram equalization[8] is particularly suited to process uniform illumination image. Although traditional edge detection is less sensitive to brightness of image, they are so sensitive to texture details which result in too much noise, while threshold extraction can, in turn, eliminate the interference of detail information, but too sensitive to unsteady light source, resulting halo phenomenon. Fourier transformation and wavelet transformation filtering are good at compression and denoising images, but it is difficult to use alone. Gulick al. developed a rock detector with cast shadows [9]. Thompson et al. developed a rock detection method based on segmentation, detection, and classification [10]. Song and Shan[11] developed a framework for automated rock segmentation from Mars rover imagery. Li et al. extracted large rocks from three-dimensional ground points generated from stereo images [3]. Hence, for natural image extraction, especially fully automated extraction without human operation, so far there is no general method. However, an organic combination of current image processing method can overcome relatively simple natural interference and bring satisfactory results. Especially the procession of spatial natural data like Mars image is relatively not enough.

In physics, a field is one of the basic forms of particle existence. A particle that pervades its energy into physical space generates a field, and the field also acts on other particles simultaneously [12]. In the context, all the fields from different local particles are superposed in the global physical space. And the superposition enables the field to characterize the interaction among different particles. Inspired by Field Theory in physics, Wang et al. [13] proposed data field to describe the interaction among data objects. Similar to physics, each data object is treated as a particle with certain mass and pervades its data energy to the whole data field in order to demonstrate its existence and action in the tasks of Mars image extraction [14][15].

In order to get more precise result, a novel method is proposed in this paper to automatically extract rocks from Martian surface images. The rest of this paper is organized as follows. Section 2 is the principles. A case is studied on the images acquired by the Spirit rover of the Mars Exploration in Section 3. Section 4 draws a conclusion.

2 Data Field

Data field is given to express the power of an object in the universe of discourse on Mars images by means of potential function as the physical field does. In data field, mutual effects among data items are indicated by a field strength function, which

might take different forms, such as nuclear form and gravitational form. At one position, the effects from different sources mutually overlay, and the superposed result is named as potential value. Given the same parameters, the potential values in densely distributed regions are much higher than in sparse regions. With the definition of an appropriate field strength function, the potential value could well reflect the data distribution. By analyzing the changing regularity of potential value, the distribution of data objects is thus explored.

2.1 Mathematical Model

In a data space $\Omega \in R^p$, a data object that is characterized by using a feature vector $x_i=(x_{i1}, x_{i2}, \dots, x_{ip})^T$, brings about a virtual field with the mass m_i . If there exists an arbitrary point $x=(x_1, x_2, \dots, x_p)^T$ in the data space Ω , the scalar field strength on an arbitrary point x of data field from x_i is defined as Equation (1).

$$\varphi(x) = m_i \times K \left(\frac{\|x - x_i\|}{\sigma} \right) \tag{1}$$

Where $m_i (\sum_{i=1}^n m_i = 1, m_i \geq 0)$ is treated as the mass of data object x_i , $K(x)$ is the unit potential function, which satisfies $\int K(x)dx = 1, \int xK(x)dx = 0$, to express the law that data object x_i always radiates its data energy in the same way in its data field. $\|x - x_i\|$ represents the distance between data object x_i and the point x in the field. $\sigma \in (0, +\infty)$ is the impact-factor-parameter that controls the interacted distance between objects. The function of σ is used to set the value of impact factor of data field. For data field, an appropriate value of impact factor could reflect the distribution of data objects rather well. In our experiment, all σ is set between 3 and 5.

Each data object x_i in dataset has its own data field in the data space Ω . All data fields in the same space are superposed and thus form the global data field. The field strength of an arbitrary point x in global data field is defined as Equation (2).

$$\varphi(x) = \sum_{i=1}^n m_i \times K \left(\frac{\|x - x_i\|}{\sigma} \right) \tag{2}$$

Where n is the number of data objects in the dataset, and other parameters take the same meaning with Equation (1).

Date field strength may reflect the distribution of data objects in data space. With the same parameters, data field strength is high in densely populated regions whereas low in sparse areas. Distribution of data items could be explored by analyzing the changing regularity of data field strength. Data field strength is a function of feature vector x . Thus, the change rate of field strength could be described by the first-order partial derivative of field strength, which is defined as Equation (3), where all parameters take the same meaning with Equation (1).

$$F(x) = \nabla \varphi(x) = \sum_{i=1}^n (x - x_i) \times m_i \times K\left(\frac{\|x - x_i\|}{\sigma}\right) \tag{3}$$

2.2 Mars Image Field

Let $A_{r \times s}$ denote an image of Mars surface based on $r \times s$ pixel. Each pixel is considered as a data object in two-dimensional generalized data field and m_{ij} as the mass of data object has been normalized to $[0, 1]$, where $m_{ij}=A(i,j)$ from the graylevel of Mars surface with $i = 1, 2, \dots, r; j = 1, 2, \dots, s$ [10]. The data field of Mars image is defined by the interaction of all the pixels in a two-dimensional image space. In order to calculate conveniently, let data set $D = \{X_1, X_2, \dots, X_n\}$ denote the pixel coordinates where $n=r \times s$ and $X_i = (X_{i1}, X_{i2})^T$. X_{i1}, X_{i2} mean horizontal coordinate and vertical coordinate respectively, such as $X_1 = (1, 1)^T, X_2 = (1, 2)^T, \dots, X_n = (r, s)^T$. The Mars surface image matrix $A_{r \times s}$ is denoted as a vector M , where $M = (m_1, m_2, \dots, m_n)^T$. According to the Equation of generalized potential function estimation [1], the potential function estimation of facial image can be calculated as

$$\varphi(x) = \sum_{i=1}^n m_i \times \left\{ \prod_{j=1}^2 K\left(\frac{x_j - X_{ij}}{\sigma_j}\right) \right\} \tag{4}$$

Where σ_1 is the impact factor of the 1 -st dimension on horizontal coordinate, and σ_2 is the impact factor of the 2 -nd dimension on vertical coordinate. As the horizontal coordinate and the vertical coordinate have the same scale, σ_1 and σ_2 are exactly equal to each other.

For different physical fields, the field strengths are mathematically modeled in various functions. According to the attenuation of field strength, physical fields could be categorized into two groups, long-range and short-range. In long-range fields, field strength function diminishes slowly and varies gently according to distance parameter. Electromagnetic field and gravitation field are two kinds of long-range fields. In short-range fields, field strength function diminishes quickly and varies rapidly on distance, for example, nuclear field.

Given appropriate parameters, nuclear field can describe the distribution of data objects inside datasets in a way that maximizes the similarity between two nearby data objects, whereas minimizes the similarity for remote objects. The field strength function defined in this paper may take the form of short-range, nuclear indeed, field strength function. Thus, the $K(x)$ in Equation (1) could be defined in the exponent form, as Equation (4), where all parameters take the same meaning with Equation (1).

$$K(x) = e^{-\left(\frac{\|x - x_i\|}{\sigma}\right)^2} \tag{5}$$

3 Principles

First, foreground rocks are differed from background information by binarizing image on rough partitioning image. Second, foreground rocks is grouped into clusters by locating the centers and edges of clusters in data field via hierarchical grids. Third, the target rocks are discovered for the MER to keep healthy paths. Main steps are shown in the algorithms.

Algorithm : Extracting rocks from Mars images

Input: images, σ , t

Output: target rocks

Process:

```

1 // Binarize image on rough grids
2 Transform an original image into a narrowed image of 256×256 by
  using average filter
3 Partition the Transformed image into some segments
4 If it is bigger than The average of the pixels in each segment, the value
   $\rho_{ij}$  of the pixel is set to be 255 in every segment
5 Take the coordinates  $(i, j)$  and the pixel mass  $m_{ij}=255-\rho_{ij}$  as the features
  of a pixel
6 //Initialize grids
7 For (data object  $o$ : all data objects
8   For  $(i=0; i<d; i++)$ 
9      $j=[f_i/s_i]$ ; //  $f_i$  is the feature value of  $o$ , while  $s_i$  is the size of grid
      on dimension  $i$ ;
10    grids $[i, j].add(o)$  ;
11    End for
12  End for
13  $\sigma = \max_{1 \leq i \leq d} s_i \times \sigma$ ;
14 For (grid  $M$ : all grids)
15   For  $(i=0; i<d; i++)$ 
16      $Fvalue = \sum o_v / M.sizeofData()$ ;
17      $M.setFeatureValue(i, fvalue)$ ; //  $o_v$  is the feature value of
      data object  $o$ , which belongs to  $M$ , on dimension  $i$ ;
18   End for
19 End for
20 //calculate first-order partial derivative potential value
21 For (grid  $m_1$ : all grids)
22   For (grid  $m_2$ : all grids)
23      $fdpValue += getFDpValue(m_1, m_2, \sigma)$ ; //calculate first-order
      partial derivative potential value while  $m_2$  is the energy source
24   End for
25    $m_1.setFdpValue(fdpValue)$ ;
26 End for
27 //identify clustering clusters
28 For (grid  $m_i$ : all grids)

```

```

29         For (i=0;i<d;i++)
30             if(MFPij-1>0&& MFPij+1<0)
31                 ccenters[i].add(mij);
32         // MFPij-1 and MFPij+1 are the first-order partial derivative potential
           value of two neighbors of M on dimension i;
33         End for
34     End for
35     filterCenters(ccenters); //detect edges and mark all densely distributed
           grids in lookupTable
36     For (Clustering Center c:ccenters)
37         For(i=0;i<d;i++)
38             j=c.getCoord(i);
39             While (--j>0)
40                 if(MFPij< MFPij+1)
41                     break;
42             else
43                 lookupTable.mark(i,j);
44             End while
45             j=c.getCoord(i);
46             While (++j<k)
47                 if(MFPij-1< MFPij)
48                     break;
49             else
50                 lookupTable.mark(i,j);
51             End while
52         End for
53     End for
54     //find full clusters from the lookupTable
55     searchFullClusters(lookupTable);
56     //discover the target rocks
57     discoverTargetRock(fullClusters);
58     Position with active contour model.

```

3.1 Differ Foreground Rocks from Background Information

It is preprocessing to binarize Mars image by partitioning the image into rough partitioning images for rapidly differing foreground rocks from background information.

First of all, an original image is transformed into a narrowed image of 256×256 by using average filter [15] for improving the algorithm performance. Then, the average of rough partitioning image is proposed to binarize Mars images in which the whole image is completely partitioned into some segments. The average of the pixels in each segment is the threshold. If it is bigger than the threshold, the value of the pixel is set to be 255 (white color) in every segment. Third, the coordinates (i, j) and the graylevel ρ_{ij} are take as the features of a pixel when the pixel value is smaller than

255. Due to the small value of Mars image rocks, the feature value of pixel greylevel is further transformed via $m_{ij}=255-\rho_{ij}$.

3.2 Initial Grids

In order to simplify the calculation, the first step is to quantize the feature space, where each dimension i in the d -dimensional feature space will be divided into k intervals. Grid-number k is used to quantize the original feature space. The quantized grids determined by k are the base of rest steps. Use these grids to calculate first-order partial derivative potential value and search clusters. The larger k is, the more accurate result will achieve. However, the intention of quantizing grids aims at accelerating; grid-number with a too large value will ruin this attempt. The assignment of grid-number parameter k needs to be resolved according to the actual distribution of data objects. Data objects will only affects objects nearby inside the area with a radius of 1.25σ . To improve the efficiency, the quantized grids could be merged to bigger grids with side length about 1.25σ , when initializing grids. This merging step can accelerate the calculation of first-order partial derivative potential value to a large extent. In most situations, the selection of k and σ will affect each other. Users can first select an appropriate impact factor which can best reflect the distribution of data objects. Set the value of k on the interaction of clusters. Then all data objects are assigned into these k^d grids, and the quantity m of data objects are recorded in each grid.

Let $F_k=(f_1, f_2, \dots, f_d)$ be the feature vector of the original data object o in the original feature space. Let $G=(v_1, v_2, \dots, v_d)$ denotes a grid in the original feature space where $v_i (1 \leq v_i \leq k, 1 \leq i \leq d)$, is the location of the unit in dimension i of the feature space. Let s_i be the size of each grid in dimension i . Data object o will be assigned to grid G if $\forall i, (v_i - 1)s_i \leq f_i < v_i s_i (1 \leq i \leq d)$. After all the data objects are assigned into grids, set the impact factor of data field, which is represented as σ . Let σ be the impact factor parameter. The impact factor σ of data field would be set according to Equation (6).

$$\sigma = \max_{1 \leq i \leq d} s_i \times ifp \tag{6}$$

The initialization of impact factor σ is followed by the calculation of first-order partial derivative potential value that is defined in Equation (3). During the calculation, original data objects would be represented by the k^d quantized grids. Each grid is taken as a new data object, represented as no . For data no_j (corresponds to grid G_j), its mass m_j equals to the quantity of data objects belonging to that grid. Let $NF_j=(nf_{j1}, nf_{j2}, \dots, nf_{jd})$ be the feature vector of no_j , nf_{ji} can be calculated as

$$nf_{ji} = \frac{\sum ov_i}{m_j} \tag{7}$$

Where ov_i represents the feature value of original data object o , which belongs to grid m_j , in dimension i .

After all the data objects are assigned into corresponding grids, the quantized grids can be regarded as data objects *no* with the mass m , which equals to the quantity of data objects in that grid. The further steps, including the calculation of potential value, are all based on the quantized grids.

3.3 Identify Clustering Centers

A cluster corresponds to a densely distributed region in feature space, whereas the clustering center is the most representative point. As demonstrated in Equation (2), potential value is proportional to the distributed density in space, and thus the center of a cluster is naturally to be the point that possesses maximum potential value. Figure 1 is the distribution of data objects in a 2-D dataset used in [16] with 10000 2-D data objects, most of which agglomerate as 9 clusters. Figure 2 is the change curve of potential value and first-order partial derivative potential value for spots in Figure 1 along the black line from left to right. The change curve of potential value (solid line) in Figure 2 demonstrates the fact that the data distribution is well reflected by data field. Each densely distributed area along the black line corresponds to a prominence in the change curve of potential value in Figure 2. Thus the task of detecting densely distributed areas could be substituted by detecting prominences along the change curve of potential value. By searching the points that possess local maximum potential value, the clustering centers could be easily found out.

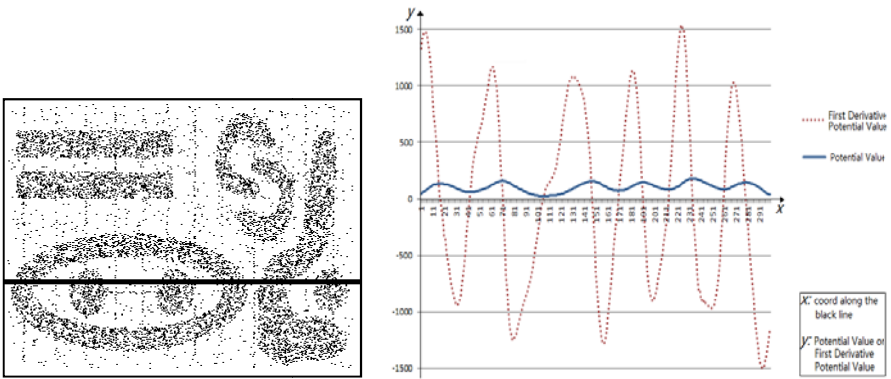


Fig. 1. Distribution of 2-D dataset **Fig. 2.** Potential curve and its first-order partial derivative

As the above mentioned, a clustering center is the point that possesses local maximum potential value. Thus, the first-order partial derivative potential value on all dimensions of a clustering center equals 0. By finding out candidate clustering centers in all dimensions and making intersection, those fake clustering centers, which refer to locations possess local maximum potential value in parts of dimensions, will be filtered. For the actual position of clustering center does no help to the

effectiveness of clustering result, use clustering grid instead. A center grid is the grid which contains clustering centers inside.

Let MFP_{ij} be the first-order partial derivative value of grid m_j in dimension i at place where NF_j locates. Grid m_j is considered to be a clustering grid only if $MFP_{ij-1} > 0$ and $MFP_{ij+1} < 0 (1 \leq j < k-1)$. A candidate center grid may only be a real center grid if it exists to be candidate in all dimensions.

3.4 Detect Edges

The density of data objects reduces from the center of cluster to its edge, and the reduction becomes tempestuous at the edge of cluster. In order to locate the edge, analyze the change rate of potential value by calculating the first-order partial derivative potential value on each dimension of the feature space. As for the dataset displayed in Figure 1, the change of potential value along the black line could be described by first-order partial derivative potential value on that dimension, which is represented as the dotted curve in Figure 2. The maximum (or minimum) points of the change curve correspond to the edge of cluster. The change of potential value in a dimension could be described by the first-order partial derivative function of potential value on that dimension.

Let $Center=(v_1, v_2, \dots, v_d)$ be one of the clustering centers, where $v_i (1 \leq v_i \leq k, 1 \leq i \leq d)$, is the location of the grid in dimension i of the feature space. Search for the neighborhood of $Center$ in each dimension, and mark such all grids as m_j or m_{j+1} that satisfy $MFP_{ij} \geq MFP_{ij+1}$, in which m_j or m_{j+1} has already been marked.

3.5 Find Full Clusters

The whole cluster is defined as the data collection whose items are surrounded by the edge, and represented by the centers. In actual cases, one cluster might be represented by more than one clustering centers, for example, ring-like shaped clusters possess more than one clustering centers along the ring. With this definition, the main steps are motivated. Firstly, identify clustering centers which possess local maximum potential value by locating those spots whose first-order partial derivative potential value equals 0. Secondly, search the neighborhood of each clustering centers, and detect the full edge by analyzing the first-order partial derivative potential value.

Let $den_i (1 \leq i \leq k^d)$ be the quantity of original data objects belong to each marked grid. Use function $f_{[\min(den_i)]}$, which is optional, to calculate the noise thread, and filter all the candidate grids whose quantity of original data objects are less than the thread.

After the filtering step, all connected candidate grids correspond to a cluster in the original data space. Flood-Fill algorithm [15] is used to find out the final clusters.

3.6 Discover Target Rocks

Automatically extracting rocks from Marts images is valuable for hazard avoidance and rover localization in rover missions. Actually, big rocks are one of the major obstacles for rover traversing and even endanger the rovers' safety, while Mars

Exploration Rover (MER) may overcome much smaller rocks. Thus the target rocks are discovered for keeping healthy paths

4 Experiments and Comparisons

The experiment is with tens of navigation cameras (Navcam) and Panoramic cameras (Pancam) images taken by MER Spirit rover and generally obtained satisfactory results. In the experiments, set the value of grid-number between 15σ for less than 30 clusters. The impact-factor parameter σ is set between 4. To improve the efficiency, the quantized grids could be merged to bigger grids with side length about 1.25σ when initializing grids. Fig.3 is original images. Fig.4, Fig.5 and Fig.6 show some representative results on Navcam images.

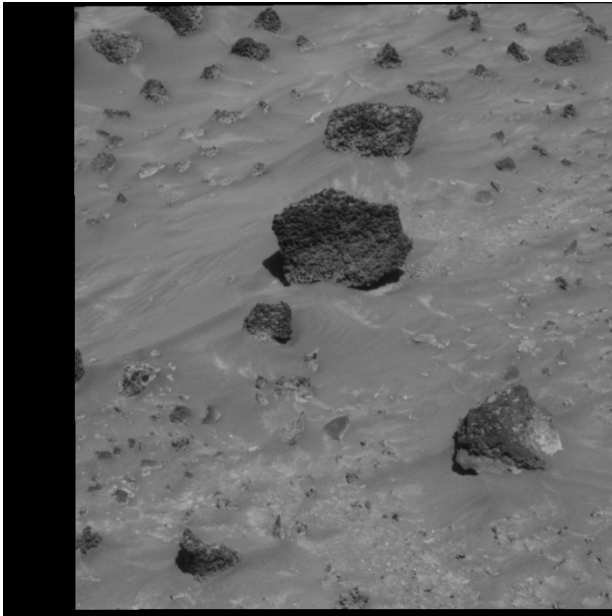


Fig. 3. Original image (Image ID of 2P207058850FFLAS00P2274L2M1.img)

Fig.3 is from 2P207058850FFLAS00P2274L2M1.img that is the left image of a stereo pair taken by Spirit rover's Navcam. Among the imaging sensors of the MER rovers, Navcam is a pair of monochrome stereo cameras mounted on the camera bar of the rover, and Pancam is a pair of multispectral stereo cameras mounted on the same camera bar. The epipolar-resampled Navcam and Pancam images, which were automatically generated by Multimission Image Processing Laboratory (MIPL) of Jet Propulsion Laboratory (JPL) with its software pipeline, are directly downloaded from

the website of MER Analyst's Notebook (<http://anserver1.eprsl.wustl.edu/>). The images are published in pds format and defined as FFL files.

In Fig.4, Fig.4(a) is the narrowed image of 256×256 on Fig.3 by using average filter [15]. Fig.4(b) denotes the binarized image after Fig.4(a) is completely partitioned into 3×3 rough segments.

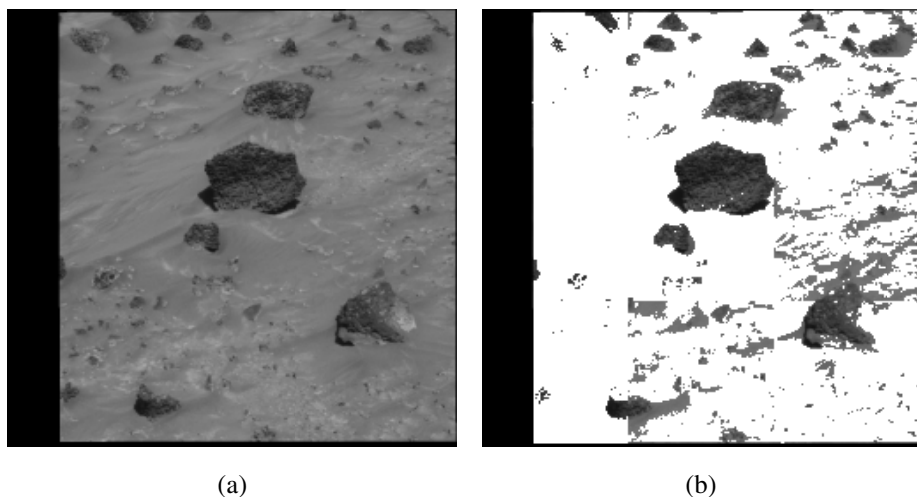
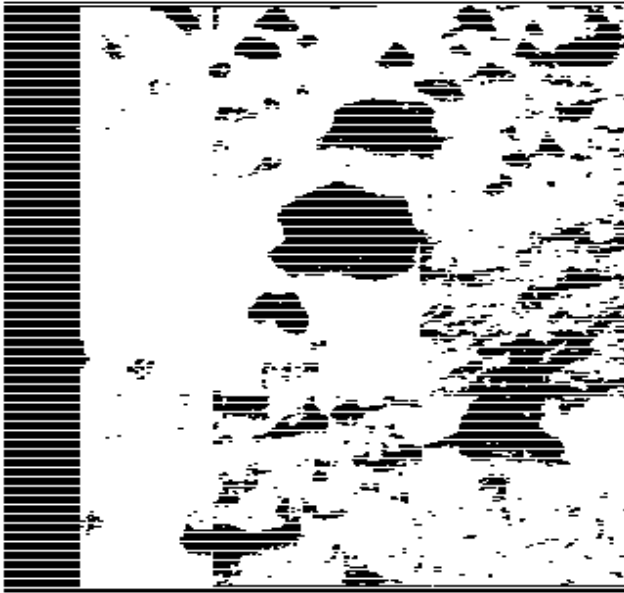


Fig. 4. Binarizing image (a) narrowed image (b) binarized image

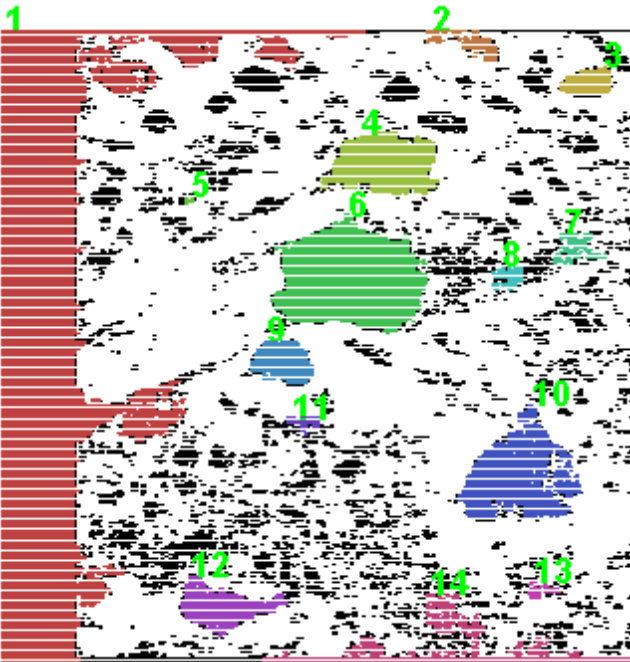
Fig. 5 is the clustering rocks on features distribution. Fig.5(a) is the distribution of the features of a pixel in Fig.4(b). That is, the further transformed values of pixel greylevel $m_{ij}=255-\rho_{ij}$. Distributes along with the coordinates (i, j) in Fig.4(b). Fig.5(b) is the resulting rock clusters, in which 14 clusters are obtained. Obviously, cluster 1 is noisy cluster. Compare Fig.5(a) and Fig.5(b), minor rocks are ignored in the result. In contrast to Fig.4(a), Fig.5(b) extracts almost all rocks. It shows that most rocks significant to scientific study have been successfully extracted, indicating that the overall performance of the proposed rock extraction method is satisfactory. In some areas where rocks are occluded or stuck together, two rocks are extracted as one rock, for example, cluster 3. In areas with large slopes, a number of rocks are partially extracted, e.g. cluster 12. These should be further investigated for improvement of the rock extraction method.

When the Mars Exploration Rover (MER) walks on the surface of Mars, big rocks are paid more attention in advance than small rocks, especially to the region in the target direction of MER walking. Fig.6 further uncovers the relatively big rocks in the target region.

In a sum, the results show that the method can extract rocks from Mars images automatically and practically, and it has potential applications.



(a)



(b)

Fig. 5. Rock clusters on features distribution (a) features distribution (d) resulting rock clusters

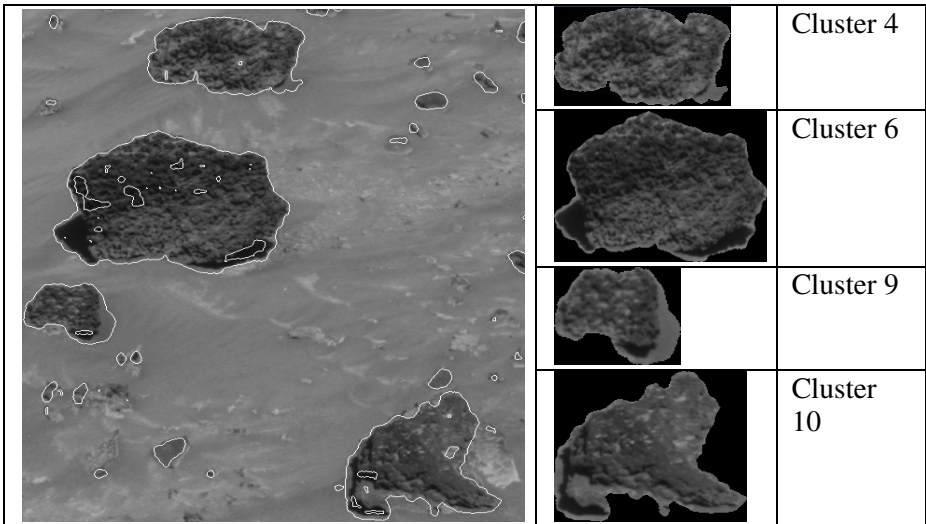


Fig. 6. Discovered rocks in the target region

5 Conclusions

Base on data field, a new method was proposed to extract rocks from Mars images. Foreground rocks were firstly differed from background information by binarizing image on rough partitioning images. Then, foreground rocks is clustered by locating the centers and edges of clusters in data field via hierarchical grids. Finally, the target rocks are discovered for the Mars Exploration Rover (MER) to keep healthy paths. Experiment results using Spirit rover data demonstrated the effectiveness of the method.

However, Mars images have specific characteristics, i.e. uneven illumination, low contrast between foreground and background, many noises in background, and rocks with irregular shape. For example, because of the effect of illumination, the shadow of stone can easily be regarded as real stone incorrectly. And in challenging areas with large slopes or where rocks stuck together, the results might be unsatisfactory. The next step is to make further texture analysis of the rocks extracted from Mars image, as well as enhancing the performance.

Acknowledgments. This paper is supported by the National 973 Program of China(2007CB310804), and the National Natural Science Foundation of China (61173061).

References

- [1] Thompson, D.R., Castano, R.: Performance comparison of rock detection algorithms for autonomous planetary geology. In: Aerospace, IEEEAC Paper No. #1251. IEEE, USA (2007)

- [2] Wagstaff, K.L., et al.: Science-based region-of-interest image compression. In: 35th Lunar and Planetary Science Conference, League City, Texas, USA (2004)
- [3] Li, R., et al.: Rock modeling and matching for autonomous long-range Mars rover localization. *Journal of Field Robotics* 24(3), 187–203 (2007)
- [4] Gor, V., et al.: Autonomous rock detection for mars terrain. In: Space 2001. American Institute of Aeronautics and Astronautics, Albuquerque (2001)
- [5] Adelman, H.G.: Butterworth equations for homomorphic filtering of images. *Computers in Biology and Medicine* 28(2), 169–181 (1998)
- [6] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
- [7] Ji, L., Yan, H.: Attractable snakes based on the greedy algorithm for contour extraction. *Pattern Recognition* 35(4), 791–806 (2002)
- [8] Yuen, P.C., Feng, G.C., Zhou, J.P.: A contour detection method: initialization and contour model. *Pattern Recognition Letters* 20(2), 141–148 (1999)
- [9] Gulick, V.C., et al.: Autonomous image analyses during the 1999 Marsokhod rover field test. *Journal of Geophysical Research* 106(E4), 7745–7763 (2001)
- [10] Thompson, D.R., et al.: Data mining during rover traverse: from images to geologic signatures. In: 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, USA (2005)
- [11] Song, Y.H., Shan, J.: Automated rock segmentation for Mars Exploration Rover imagery. In: Lunar and Planetary Science Conference XXXIX, Houston, USA (2008)
- [12] Giachetta, G., Mangiarotti, L., Sardanashvily, G.: *Advanced Classical Field Theory*. World Scientific Publishing Co. Pte. Ltd (2009)
- [13] Wang, S.L., Gan, W.Y., Li, D.Y., Li, D.R.: Data Field for Hierarchical Clustering. *International Journal of Data Warehousing and Mining* 7(4), 43–63 (2011)
- [14] Li, D.R., Wang, S.L., Li, D.Y.: *Spatial Data Mining theories and applications*. Science Press, Beijing (2006)
- [15] Gonzalez, R.C., Woods, R.E.: *Digital image processing*, 3rd edn. Pearson Education, Upper Saddle River (2008)
- [16] Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical Clustering Using Dynamic Modeling. *Computer* 32(8), 68–75 (1999)

Finding a Wise Group of Experts in Social Networks

Hongzhi Yin, Bin Cui, and Yuxin Huang

Department of Computer Science & Key Lab of
High Confidence Software Technologies (Ministry of Education),
Peking University, 100871 Beijing, China
{bestzhi, bin.cui, huangyuxin}@pku.edu.cn

Abstract. Given a task T , a pool of experts χ with different skills, and a social network G that captures social relationships and various interactions among these experts, we study the problem of finding a wise group of experts χ' , a subset of χ , to perform the task. We call this the Expert Group Formation problem in this paper. In order to reduce various potential social influence among team members and avoid following the crowd, we require that the members of χ' not only meet the skill requirements of the task, but also be diverse. To quantify the diversity of a group of experts, we propose one metric based on the social influence incurred by the subgraph in G that only involves χ' . We analyze the problem of Diverse Expert Group Formation and show that it is NP-hard. We explore its connections with existing combinatorial problems and propose novel algorithms for its approximation solution. To the best of our knowledge, this is the first work to study diversity in the social graph and facilitate its effect in the Expert Group Formation problem. We conduct extensive experiments on the DBLP dataset and the experimental results show that our framework works well in practice and gives useful and intuitive results.

Keywords: team formation, social influence, social network, heuristics algorithm.

1 Introduction

The credibility of a group of experts depends not only on the expertise of the people who are involved, but also on whether they can give full play to their professional expertise and make a wise and fair decision. Diversity plays an important role in the process of decision-making of an expert group, and lack of diversity may result in potential conflict of interest (COI) [1] and even bring about the situation where experts begin ignoring their own expertise and following the crowd due to social pressure to conform or information cascade effect [4].

In the book of *Wisdom of the Crowds*, James Surowiecki, the business columnist for *The New Yorker*, asserts that “under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them.” In his opening example, Surowiecki notes that if many people are guessing independently, then the average of their guesses is often a surprisingly good estimate of whatever they are guessing about (perhaps the number of jelly beans in a jar, or the weight of a bull at a fair).

The key to this argument of course is that the individuals each utilize their own information (their signals), and they guess independently, without social influence from others, and do not know what the others have guessed. If, instead, they are close friends and can know or observe the earlier guesses of others, then they may tip into the setting where they follow the crowd due to social pressure to conform or information cascade effect, and there would be no reason to expect the average guess to be good at all [4].

For example, consider a hiring committee that needs to decide whether to make a job offer to candidate A or candidate B. In these kinds of situations, a common strategy is to go around the table, asking people in sequence to express their support for option A or option B. But if the participants assume that they all have roughly equal insight into the problem, then a cascade can quickly develop: if a few people initially favor A, others may be led to conclude that they should favor A, even if they initially preferred B on their own. According to Surowiecki's arguments in his book, we can conclude that the "right circumstances for an expert group to make a credible decision" consists of four basic conditions, which are that "wise groups" are effective when they're composed of individuals who have diverse opinions; when the individuals aren't constrained and are free to express their opinions; when there's diversity in the crowd; and when there's a way to aggregate all the information and use it in the decision-making process.

We will next take scientific peer-review as an example to illustrate what constitutes an ideal expert group. A wise group of diverse reviewers are expected in the review assignment in order to reduce social influence among team members and avoid their biased evaluation. We hope that each expert in the expert group can make his or her own decision, freely and independently. Assume, for example, a program director in NSF who wants to find a group of reviewers to review a proposal q written by a , covering the following topics: $q = \{algorithm, data\ mining, information\ retrieval, web\ service\}$. Also assume there are seven reviewer candidates, $\{a, b, c, d, e, f, g\}$, with the following expertise:

- $X_a = \{algorithm, data\ mining, information\ retrieval, web\ service\}$
- $X_b = \{data\ mining, machine\ learning, algorithm\}$
- $X_c = \{web\ programming, web\ service\}$
- $X_d = \{data\ mining, information\ retrieval, algorithm\}$
- $X_e = \{web\ service, web\ programming\}$
- $X_f = \{data\ mining, information\ retrieval, algorithm\}$
- $X_g = \{web\ service, web\ programming\}$

The relationship among these candidates are illustrated in the social network graph G shown in Figure 1, where the existence of an edge between two nodes in G indicates that there exists some social or professional relationship between the two corresponding researchers. Without considering conflict of interest, the program director can select either $\chi_1 = \{a\}$, $\chi_2 = \{d, c\}$, $\chi_3 = \{d, e\}$, $\chi_4 = \{f, e\}$, $\chi_5 = \{f, c\}$, $\chi_6 = \{f, g\}$ or $\chi_7 = \{d, g\}$. Each of these groups can cover the required expertise. There are four solutions χ_3, χ_4, χ_6 and χ_7 left when considering the social tie between reviewer candidate c and author a , and the existence of graph G makes χ_4 and χ_7 two "wise groups" when taking the social influence among reviewer candidates into account, since the structure of G indicates that there is not any social tie between d and g , e and f ; thus, they can be free to express their opinions, not affected by each other.

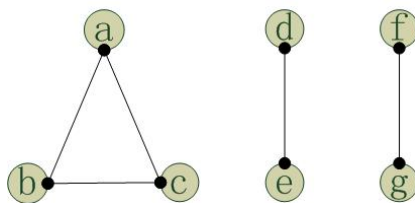


Fig. 1. Network of connections between researchers in $\{a,b,c,d,e,f,g\}$

The existence of a social network between individuals is quite common in real scenarios. In a geographical map of experts' distribution over research organizations and universities, the graph encodes the fact that experts in the same research organizations or universities have close social distance, and researchers in different cities have longer distance than those in the same city. In a research community, the network captures previous successful collaborations among researchers. Other examples of social networks between professionals include LinkedIn(www.linkedin.com) which comprises a large number of people from information technology areas. When people are connected by a network, it becomes possible for them to influence each other's behavior and decisions [4,5].

The problem: In this paper, we study the problem of finding a group of individuals who can function as a wise and creditable expert group to accomplish a specific task (e.g. scientific peer-review). We assume that there exists a pool of n expert candidates $\chi = \{1, \dots, n\}$ in a social network G , where each candidate i has a set of skills X_i . Given a task T that requires a set of skills, our goal is to find a group of experts $\chi' \subseteq \chi$, such that every required skill in T is exhibited by at least one expert in χ' . Moreover, the members of χ' should be diverse enough to reduce social influence among group members. The diversity means and measures how independently, rationally the group members can utilize their expertise to make a credential and fair decision: the more diverse the group members are, the better quality of the expert group.

In summary, our contributions are as follows:

- To the best of our knowledge, we are the first to study diversity in the presence of social networks and utilize it in the Expert Group Formation.
- We propose one novel formulation to compute social influence, based on which we define the metric to quantify the diversity of a group of experts.
- We study and analyze the problem of Diverse Expert Group Formation rigorously and propose two appropriate approximation algorithms for the problem.
- We conduct extensive experiments and the experiments illustrate that our problem definition, as well as our algorithms, work well in practice and give useful and intuitive results.

The rest of the paper is organized as follows: in Section 2 we formally define the Diverse Expert Group Formation problem. In Section 3 we propose one novel formulation to compute social influence. In Section 4 we propose two approximation algorithms for the

Diverse Expert Group Formation problem and in Section 5 we illustrate the usefulness of our methodology on a real review-assignment dataset. In Section 6 we review the related work on Group Formation problem and we conclude the paper in Section 7.

2 Problems

This section first present necessary preliminaries and then formally define the problem.

2.1 Preliminaries

Given a pool of candidates consisting of n experts, $\chi = \{1, \dots, n\}$, and a universe of m skills $\mathcal{A} = \{a_1, \dots, a_m\}$. Each expert i is associated with a set of skills $X_i \subseteq \mathcal{A}$. If $a_i \in X_i$ we say that expert i has expertise a_j ; otherwise expert i does not have expertise a_j . We often use the set of skills an expert possesses to refer to him. Also, we say that a group of experts $\chi' \subseteq \chi$ possess skill a_j if there is at least one expert in χ' that has a_j .

A task T is simply a subset of skills and expertise required to perform a job, e.g. to make a big decision. That is, $T \subseteq \mathcal{A}$. If $a_j \in T$ we say that skill a_j is required by task T . Next, we define the *cover* of a group of experts χ' with respect to task T , denoted by $C(\chi', T)$, to be the set of skills that are required by T and for which there exists at least one expert in χ' that posses them. That is, $C(\chi', T) = T \cap (\cup_{i \in \chi'} X_i)$. Similar to inverted index, given a skill $a \in \mathcal{A}$, we define its support set, denoted by $S(a)$, to be the set of experts in χ that has this skill. That is, $S(a) = \{i | i \in \chi \wedge a \in X_i\}$.

For every two nodes $i, i' \in \chi$, we take $Pr_i(i')$ to represent the value of social influence that node i receives from node i' . It should be noted that this metric is asymmetric. To quantify the diversity between two nodes in the social graph G , we define a social influence-based diversity metric as follows:

$$SI(i, i') = \max\{Pr_i(i'), Pr_{i'}(i)\} \quad (1)$$

where we transform the asymmetric social influence metric to the symmetric social influence metric, which lays foundations to compute social influence-based diversity for a group of nodes χ' .

To measure the diversity of a group of experts χ' , we extend the above diversity metric between two nodes to social influence-based diversity metric among more expert nodes, as follows:

$$SI(\chi') = \arg_{i, i' \in \chi'} \max SI(i, i') \quad (2)$$

2.2 Problem Definition

In this subsection, we formally define the problem of Expert Group Formation. Our problem definition reflect our belief that less social influence among group members is an important factor for the successful completion of a task such as making a credential and wise decision.

Expert Group Formation Problem : Given the set of n experts $\chi = \{1, \dots, n\}$, a social graph $G(\chi, E)$, and task T , find a group of experts $\chi' \subseteq \chi$, so that $C(\chi', T) = T$, and the social influence $SI(\chi')$ is minimized.

Theorem 1. *The Group Formation problem is NP-complete.*

Proof. We prove the proposition by giving a simple reduction to the MULTIPLE CHOICE COVER(MCC) problem. It is well known that the MCC problem is NP-Hard as shown in [2]. In the decision version of the MCC problem, we are given a universe $V = \{1, \dots, N\}$ of N elements, a $N \times N$ real matrix D with non-negative entries, and a $S = \{S_1, \dots, S_k\}$ such that each $S_i \subseteq V$. Given a constant K , we are asked whether there exists $V' \subseteq V$ such that for every $i \in \{1, \dots, k\}$, $|V' \cap S_i| > 0$ and $\max_{(u,v) \in V' \times V'} D(u, v) < K$.

We transform an instance of the MCC problem to an instance of the Group Formation problem as follows: for every set S_i in the MCC problem we create a skill a_i . The task T to be performed requires all the k skills. That is, $T = \{a_1, \dots, a_k\}$. For every element $v \in V$ of the MCC instance, we create an expert i_v with expertise $X_v = \{a_i | v \in S_i\}$. Two experts i_v and i'_v are connected in the graph G with the social influence $SI(i_v, i'_v)$.

Given this mapping it is easy to show that there exists a solution to the MCC problem with the cost at most K if and only if there exists a solution to the Group Formation problem with social influence at most K . The problem is trivially in NP.

According to the above Theorem, we can conclude that Diverse Expert Group Formation problem is NP-complete.

2.3 Discussion

In the definition of the Expert Group Formation problem and its specializations, we focused on minimizing the social influence among group members in order to maximize the diversity of the expert group. Other notions of the “effectiveness” of a group can lead to different optimization functions. Authors in [8] focus on minimizing the communication cost among group members, and the traditional Team Formation problem aims to find $\chi' \subseteq \chi$, such that $C(\chi', T) = T$ and $|\chi'|$ are minimized. The traditional problem definition ignores the existence of the underlying graph $G(\chi, E)$, and is actually an instance of the classic Set Cover problem, which can be solved by the standard GreedyCover algorithm. Details are presented in our experimental section. Although we do not study the traditional version of the problem in this paper, we note that a solution with minimum social influence implicitly requires a small group, since larger groups typically result in higher social influence.

3 Computation of Social Influence

3.1 Social Influence

When people are connected by a network, it becomes possible for them to influence each other’s behavior and decisions. Their initial circle of influence is the group of individuals that they most interact with. Because the network is a highly connected graph [10, 12], this circle of influence grows and may shape and change the thoughts of others in the group. This circle of influence also grows throughout the community. We introduce the notion of *Social Influence* $P_v(s)$ to represent the social influence and the commitment value that node v receives from node s .

Consider a social network (as shown in Figure 3) where nodes represent experts and links represent the social ties among them. Since the network captures real-world behavior, colleagues and friends in real life are likely to be linked either directly or within

two hops. Since nodes that are closer to i than others are likely to have greater social influence on i than nodes further away, a baseline approach would be to design a function to compute the social influence that node v receives from node s , with the social distance between v and s as the parameter. However, this approach is naive since it makes the assumption that all nodes within the same social distance from a target node will have the same social influence on the target node. In some cases this assumption is not true, as two nodes with the same social distance from the third node may not necessarily be impacted by the third node in the same way. For example, as shown in Figure 3 we can see that nodes a and b are at the same distance from node c . However, a is well-connected with other nodes in the same community, which results in that there are more information diffusion paths from node c to node a . One can justly argue that c should have different social influences on node a and node b , and this needs to be captured by our model. Besides, according to the widespread social and psychological phenomenon—Obedience to Authority, an expert with higher authority and status should have more social influences on others. Hence, to compute social influence in a social network, we need to concentrate on not only social distance but also *local connectivity* and Authority of effect information.

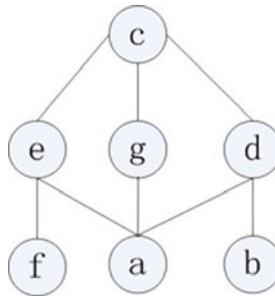


Fig. 2. Example for Node Social Influence

The above discussion leads to the following desiderata for computing social influence.

1. **Inverse Social Distance weighting:** The social influence that a node receives from the source node should be inversely proportional to its social distance from the source. The intuition behind this is that a node is likely to be affected more by behaviors occurring near itself than those occurring some distance away.
2. **Local Connectivity:** Nodes that are well-connected, i.e. having links to many other nodes within the same community, should receive more social influence from the source node.
3. **Effect of Authority:** An expert node with higher authority and status should have more social influences on others because an expert with high status and prestigious, can easily lead to others and influence others opinion. Moreover, how much social influence node v receives from node s not only depends on the authority of node s , but also depends on the authority of node v . If s is less authoritative than v , then the social influence from s to v should be damped and v does not care much about the view of s .

3.2 Approach for Computation

Precisely describing the social influences between reviewers is a tough task. One observation is that the social influence should decrease with the increasing social distance between experts in the social graph. Another observation is that the social influence should increase with the increasing local connectivity. In this part, we first propose local betweenness to measure local connectivity, and then use an exponential decaying function to simulate the propagation of social influence in the social network. Finally, we adopt sigmoid function to compute the effect of authority on the propagation of social influence. It should be noted that other machine learning based information diffusion [13] satisfying the above two criterias, and even Random Walk Model, can be also used to compute social influence. Due to space limitation, we do not present or discuss other models.

Local Betweenness: The betweenness centrality measure, which was first introduced by Freeman [11], is a global topological measure and computes, for each node in the graph, the quantity of shortest paths that pass through it. This quantity is an indicator of who the most influential people in the network are, the ones who control the flow of information between most others. The vertices with highest betweenness also result in the largest increase in typical distance between others when they are removed [11]. In our case, we are interested in the reachability of nodes in the social network from a given source node and we define it as Local Betweenness Centrality for a source node. The Local Betweenness Centrality for a given node x with respect to a source node s given a social network $N = (V, E)$ can be calculated as:

$$B(x, s) = \frac{SP_x}{|V| - 1} \tag{3}$$

where SP_x is the number of shortest paths passing through node x from source node s to other nodes in the social network. Intuitively, the nodes with high local betweenness in terms of paths from the source to other nodes should have great social influence on the source node. Authors in [11] propose two algorithms to compute betweenness centrality and their methods can be directly applied to compute Local Betweenness Centrality.

In order to satisfy the above desiderata, we propose the following formula to compute *Social Influence* $P_v(s)$ that node v receives from node s .

$$P_v(s) = \frac{1}{1 + e^{-(I(s)-I(v))}} * \lg B(s, v) * exp(-|s - v|) (v \neq s) \tag{4}$$

where $I(s)$ denotes the importance and authority of expert s , $|s - v|$ denotes the social distance between s and v in the social graph, and $B(v, s)$ is Local Betweenness of node v with respect to a source node s . This formula shows how to compute the social influence that v receives from node s , and it can be observed that the social influence $P_v(s)$ decrease with the increasing social distance and increase with the increasing local betweenness. The sigmoid function with $I(s) - I(v)$ as parameter can perfectly capture the Effect of Authority. For simplicity, we define $P_v(s) = 0$ if $v = s$.

4 Algorithms

As proved, finding an exact optimal solution for Expert Group Formation is NP-hard, so we propose two appropriate approximate algorithms, RarestFirst and Simplified RareFirst (SRareFirst) in this section.

4.1 RarestFirst Algorithm

Algorithm 1 shows the pseudocode of the RarestFirst algorithm for finding a group of diverse experts for a particular task T . As the cost of computation of social influence is expensive, we choose to pre-compute the social influence between any two experts offline, and store the results in a social influence matrix M where M_{ij} denotes $SI(i, j)$, the social influence between i and j . Physically, we use hash tables for storing the attributes of every expert (e.g. his social influence to other experts), and a different set of hash tables for storing the experts that has a specific attribute (e.g. a skill or expertise). After computation of social influence, we first identify a set of experts S_{a_i} for each skill a_i required by task T (line 4-6). Then, the algorithm sorts all expert's sets S_{a_1}, \dots, S_{a_k} incrementally order by the cardinality $|S_{a_i}|$, the number of experts with skill a_i (line 8-9). One observation is that an expert with one particular special skill or expertise required by the task should enjoy priority to be selected as the member of the expert group if only few experts have that particular skill. In the extreme situation where only one expert has one particular skill required by the task, that expert must be included in the expert group so as to perform the task. Following the observation, we start by picking one expert from S_{a_1} (line 12), and then iteratively select the next expert until all skills required by T are covered. Thus, at each step the best expert j is the one that works together with $\{1, \dots, j-1\}$ to make the social influence SI minimized (line 13-16). Among $|S_{a_1}|$ already found expert groups, we choose the expert group χ' which leads to the smallest social influence $SI(\chi')$ (line 17-19).

As one may expect, the solutions provided by our greedy algorithm RarestFirst do not have any provable quality guarantee. Thus, in Section 5 we study the greedy algorithm experimentally, and our experimental results show that in practice the RarestFirst algorithm give good solutions. From the algorithm, we can know why we set $Pr_i(i) = 0$. By defining $Pr_i(i) = 0$, experts with more skills enjoy priority to be picked as the members of the expected expert group. Meanwhile, prior choosing experts with more skills implicitly reduce the cardinality of the expert group, which potentially reduce social influence among group members. The online time required for the execution of the *RarestFirst* algorithm is $O(|S_{a_i}| \times |T| \times n)$. A worst-case analysis suggests that $|S_{a_i}| = O(n)$ and $|T| = O(n)$. Thus, the worst-case running time of the *RarestFirst* is $O(n^3)$. However, in practice, the running time of the algorithm is much less than this worst-case analysis suggests. It should be noted that this algorithm can get the optimal solution when $|T| = 2$.

4.2 Simplified RarestFirst Algorithm

Since the online time cost for the execution of the *RarestFirst* algorithm is expensive at worst case, we propose one Simplified RareFirst algorithm to reduce the required run-time at the worst case (see details in Algorithm 2). Being different from the above

Algorithm 1. The RarestFirst algorithm for finding a group of diverse experts**Input:**

A Social GRAPH, $G(\chi, E, W)$;
 The task to be performed, T ;
 Experts' skill vectors $\{X_1, \dots, X_n\}$

Output:

A credential group of experts $\chi' \in \chi$;

```

1: Initialize  $\chi'$  to be an empty set.
2: Initialize  $SI(\chi') \leftarrow \infty$ 
3: Compute social influence matrix  $M$  offline.
4: for every expertise  $a_i \in T$  do
5:    $S_{a_i} = \{i | a_i \in X_i\}$ 
6: end for
7: //Assume that  $k$  skills are required by  $T$ 
8: Set  $S = \{S_{a_1}, \dots, S_{a_k}\}$ 
9: Sort  $S$  incrementally order by  $|S_{a_i}|$ 
10: for every expert  $i \in S_{a_1}$  do
11:   Initialize  $\chi''$  to be an empty set.
12:   add  $i$  into  $\chi''$ .
13:   for every  $S_{a_j} \in S$  and  $S_{a_j} \neq S_{a_1}$  do
14:      $j \leftarrow \arg_{j \in S_{a_j}} \min SI(\chi'' \cup \{j\})$ 
15:      $\chi'' = \chi'' \cup \{j\}$ 
16:   end for
17:   if  $SI(\chi') > SI(\chi'')$  then
18:      $\chi' = \chi''$ 
19:   end if
20: end for
21: return  $\chi'$ ;

```

RareFirst algorithm, we take one heuristic rule to select the first expert in the simplified version of RareFirst Algorithm : the one with the most skills required by the task will be selected as the seed expert for the expert group formation (line 10-11). Then iteratively select the next expert until all skills required by T are covered. Thus, at each step the best expert j is the one that works together with $\{1, \dots, j-1\}$ to make the social influence SI minimized (line 12-14).

The run time required for the execution of the Simplified RarestFirst algorithm is $O(|T| \times n)$. A worst-case analysis suggests that $|T| = O(n)$. Thus, the worst-case running time of the Simplified RarestFirst is $O(n^2)$. However, in practice, the running time of the algorithm is much less than this worst-case analysis suggests.

5 Experiments

In this section, we evaluate the proposed algorithms for the Diverse Group Formation problem using the scientific-collaboration graph extracted from the DBLP dataset. We show that our algorithm for the Diverse Expert Group Formation problem give high-quality results in terms of the social influence of team, the cardinality of the team, and the minimum social distance of the team.

Algorithm 2. The Simplified RarestFirst algorithm for finding a group of diverse experts

Input:

A Social GRAPH, $G(\chi, E, W)$;
 The task to be performed, T ;
 Experts' skill vectors $\{X_1, \dots, X_n\}$

Output:

A credential group of experts $\chi' \in \chi$;
 1: Initialize χ' to be an empty set.
 2: Initialize $SI(\chi') \leftarrow \infty$
 3: Compute social influence matrix M offline.
 4: **for** every expertise $a_i \in T$ **do**
 5: $S_{a_i} = \{i | a_i \in X_i\}$
 6: **end for**
 7: //Assume that k skills are required by T
 8: Set $S = \{S_{a_1}, \dots, S_{a_k}\}$
 9: Sort S incrementally order by $|S_{a_i}|$
 10: $j = \arg_{i' \in S_{a_1}} \max C(i', T)$
 11: add j into χ'
 12: **for** every $S_{a_j} \in S$ and $S_{a_j} \neq S_{a_1}$ **do**
 13: $j \leftarrow \arg_{j' \in S_{a_j}} \min SI(\chi' \cup \{j'\})$
 14: $\chi' = \chi' \cup \{j\}$
 15: **end for**
 16: **return** χ' ;

5.1 Other Algorithms

Greedy-SIR algorithm: In this subsection, we extend the classic GreedyCover algorithm and propose Greedy Social Influence Rate algorithm. The intuition behind this algorithm is that smaller groups typically result in smaller social influence. The rationale of the algorithm is to form a solution iteratively. At round t , group χ_t is formed by adding to the group χ_{t-1} a node $i \in \chi \setminus \chi_{t-1}$. The node i is selected so that it maximizes the ratio

$$i = \arg_{i' \in \chi \setminus \chi_{t-1}} \max \frac{|C(\chi_{t-1} \cup i', T) - C(\chi_{t-1}, T)|}{SI(\chi_{t-1} \cup i') - SI(\chi_{t-1}) + \beta} \quad (5)$$

That is, the node i that achieves the best ratio of newly covered skills in T divided by the corresponding social influence is picked. The run time required for the execution of the *GreedySIR* algorithm is $\mathcal{O}(|T| \times n)$. A worst-case analysis suggests that $|T| = \mathcal{O}(n)$. Thus, the worst-case running time of the *GreedySIR* is $\mathcal{O}(n^2)$. However, in practice, the running time of the algorithm is much less than this worst-case analysis suggests.

5.2 Data Preparation

In our problem setting, we need both the social network and experts' skill sets. Since no benchmark is available for performance evaluation, we use a snapshot of the DBLP data to create a benchmark dataset for our experiments. There are 707,987 researcher

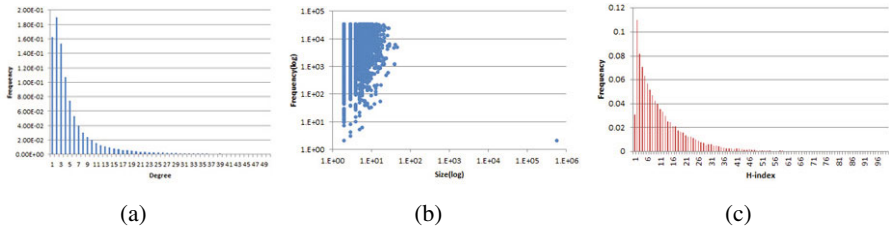


Fig. 3. (a)Nodes degree distribution; (b)Distribution of the size of the connected components; (c)Distribution of H-index of CS researchers in DBLP

nodes in the original co-author network. After removing isolated nodes, i.e., nodes that were not connected to other nodes, there are 664, 949 researcher nodes and 2, 201, 372 edges in the co-author network. In Figure 3 we report some statics about the network we generated. The co-author network is quite disconnected; it contains 32, 930 connected components, one of which has 565, 301 nodes (see the bottom-right corner of Figure 3(b)), while all the others have less than 50 nodes.

Then, we select 180 researchers from the co-author network as experts instead of taking all researchers as prospective experts, and the remaining researchers act as media nodes to spread social influence. All these 180 experts are from the largest component, and they are a subset of the reviewer candidate set created by authors in [7]. They have published at least 3 papers in conferences such as WWW, SIGIR and CIKM in order to consider only IR researches for evaluating purposes. For modeling experts’ skills, they create a profile for each expert by concatenation of all papers written by the specific expert. If a paper has more than one author, they replicate that paper, one for each author. After that, they invite an information retrieval expert to identify 25 major topics based on the topic areas in the Call for Papers of ACM WWW, SIGIR, CIKM in most recent five years and session titles in the recent conferences. The expert then reads the experts’ profiles, and assign relevant skills/topics to each expert. For example, Aristides Gionis is one of our experts, and his research topics are identified and labeled as a expertise set {T2, T9, T10} where T2, T9 and T10 represent “Clustering”, “Web IR” and “Web Structures” respectively. As shown in Figure 5(a) most experts cover 3-5 topics/skills. To model the authority and reputation of an expert, we obtain h-indexes for CS researchers in DBLP with access to the Thomson ISI Web of Science. In Figure 3(c) we present the distribution of h-numbers for all Computer Science researchers in DBLP. The average h-index of our expert set is 16.29.

5.3 Performance Evaluation

This section evaluates the Expert Group Formation algorithms on the *social influence* of the team, the *cardinality* of the team, the *minimum social influence* of the team.

Task Generation: Every generated task is characterized by one parameter: *t*- the number of required skills in the task. We use $T(t)$ to refer to a task generated for a specific configuration of the parameter *t*. Specifically, a task $T(t)$ is generated as follows: we randomly pick *t* required skills from the 25 major topics identified by the invited expert. For the results we report in this section we use $t \in \{3, 4, \dots, 20\}$. For every *t*

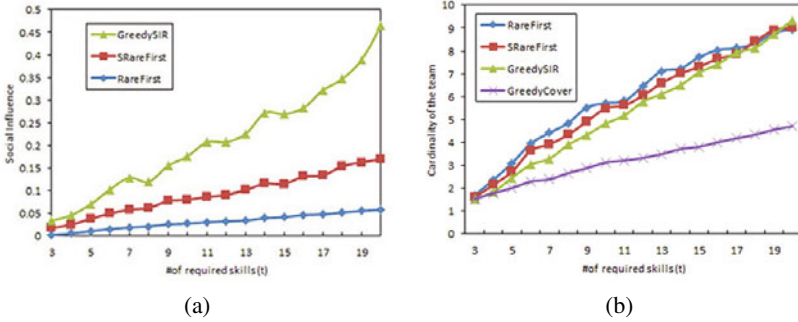


Fig. 4. (a)Average social influence of the teams produced by each Expert Group Formation algorithms for tasks $T(t)$ with $t \in \{3, 4, \dots, 20\}$; (b)Average cardinality of the teams reported by RareFirst, SRareFirst, GreedySIR, GreedyCover

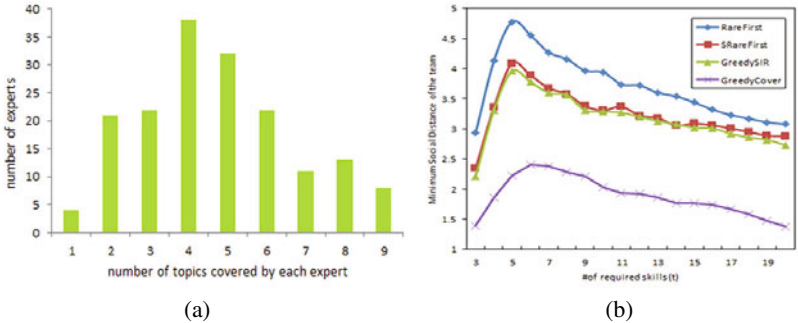


Fig. 5. (a)Distribution of the number of topics covered by each expert;(b)Average minimum social distance of teams produced by each Expert Group Formation algorithm

configuration we generate 100 random tasks for this configuration and report the average results obtained by the different methods.

Social Influence: Figure 4(a) shows the average social influence of the solutions achieved by RareFirst, Simplified RareFirst(SRareFirst) and Greedy SIR on tasks $T(t)$ with $t \in \{3, 4, \dots, 20\}$. It can be observed that , in terms of the social influence, both RareFirst and SRareFirst significantly outperforms GreedySIR; RareFirst performs better than SRareFirst. Another observation is that the social influence among team members increases with the increasing number of required skills by the task, which is consistent with our intuition and common sense. Specifically, the social influence of the teams produced by RareFirst and SRareFirst displays linear growth with the number of required skills while GreedySIR tends to exponential growth. The conclusion is that our proposed algorithms can form expert groups that are able to accomplish a given task with low social influence.

Cardinality of the team: Since the size of the team often has a positive correlation with the expense of a project, we evaluate the cardinality of the teams formed by every Expert Group Formation algorithm. The results in Figure 4(b) show that the GreedySIR slightly outperforms RareFirst and SRareFirst in terms of cardinality of the team when

the number of required skills is relatively small. For comparison, we also include the cardinality of the team reported by the GreedyCover algorithm. Recall that GreedyCover ignores the existence of the social graph and only reports a set of individuals who can perform the task by simply looking at their skillsets. Therefore, the cardinality of this solution is lower bound on the cardinality of the solutions produced by all the three aforementioned algorithms. However, since GreedyCover ignores the graph structure, it often forms such teams that there exist close social or professional ties among team members, which is most likely to result in that the teams can not make a wise and credential decision due to potential high social influence. The following experiment illustrates the validity of this claim.

Minimum Social Distance of the team: Although we take the Social Influence as the primary measure to evaluate the effectiveness of our proposed algorithms, we still report the minimum social distance among team members due to that distance is more intuitive and easy to understand. As shown in Figure 5(b), RareFirst, SRareFirst and GreedySIR significantly outperforms GreedyCover in terms of minimum social distance. The minimum distance among team members, produced by GreedyCover algorithm, is less than 2 hops in most cases. It is obvious that there are close social or professional ties among the team members in these cases.

6 Related Works

There is a considerable amount of literature on Team Formation in the operation research community [3, 6, 14]. A trend in this line of work is to formulate the Team Formation problem as an integer linear program (ILP), and then focus on finding an optimal match between people and the demanded functional requirements. The problem is often solved using techniques such as simulated annealing [3], branch-and-cut or genetic algorithms [14]. Interested readers may identify the connection between this formulation and the assignment problem. The main difference between the studies above and our work is that we explicitly take into account the social graph structure of the individuals when deciding the right group. In most of the previous work, the professional or social bonds among individuals are ignored and the focus is limited on their skills. Moreover, the problem formulation we provide, and the algorithms we propose in this paper, are fundamentally different from those proposed in the operation community literature.

The network structure between individuals in a workforce pool has been studied by authors in [9]. The authors provide an experimental study of how different graph structures among the individuals affect the performance of a team. Although related, the work presented in [9] does not address the computational problem of finding a group of experts in a given network. Authors in [8] study the problem of Team Formation, which aims to find a team of experts with lowest communication cost in social networks to complete a given task. Their problem formulation is completely different from ours.

7 Conclusions and Future Work

In this paper, we have addressed the problem of forming a team of skilled individuals to perform a given task, while minimizing the social influence among the members of the

team. We explored a formulation for the social influence among a team, which we believe are practical and intuitive. We proved that the Diverse Expert Group Formation problem is NP-hard and proposed two appropriate approximation algorithms. In a thorough experimental evaluation, we evaluated the performance of our algorithms, and compared them with reasonable baseline approaches. The experimental results show that our proposed algorithms work well in practice and gives useful and intuitive results.

In our setting, we assume that experts either have a skill or not; we do not allow for a scaling of the experts' abilities. Similar for the tasks; we assume that a task requires a certain set of skills, without considering the special importance that different skills might have for the completion of the task. However, In real world, the case is more complex. One interesting future research direction is to study the graded variant of Expert Group Formation.

Acknowledgments. This research was supported by the National Natural Science foundation of China under Grant No. 60933004, 61073019 and 61050009.

References

1. Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A.P., Arpinar, I.B., Joshi, A., Finin, T.: Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. In: WWW 2006, pp. 407–416 (2006)
2. Arkin, E.M., Hassin, R.: Minimum-diameter covering problems. *Networks* 36, 147–155 (2000)
3. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. *Cybern. Syst.* 38, 155–185 (2007)
4. David Easley, J.K.: *Networks, Crowds, and Markets Reasoning About a Highly Connected World*. Cambridge University (2010)
5. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: WSDM 2010, pp. 241–250 (2010)
6. Chen, S.j., Lin, L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering (2004)
7. Karimzadehgan, M., Zhai, C., Belford, G.: Multi-aspect expertise matching for review assignment. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 1113–1122 (2008)
8. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 467–476 (2009)
9. Gaston, J.M., des Jardins, M.: Adapting network structures for efficient team formation. In: Proceedings of the AAAI Fall Symposium on Artificial Multi-Agent Learning (2004)
10. Newman, M.E.J.: Scientific collaboration networks. i. network construction and fundamental results. *Rev. E* 64 (2001)
11. Newman, M.E.J.: Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical Review E* 64(1), 016132+ (2001)
12. Newman, M.E.J.: Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences*, 5200–5205 (2004)
13. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: KDD 2009, pp. 807–816 (2009)
14. Wi, H., Oh, S., Mun, J., Jung, M.: A team formation model based on knowledge and collaboration. *Expert Syst. Appl.*, 9121–9134 (July 2009)

Fully Utilize Feedbacks: Language Model Based Relevance Feedback in Information Retrieval

Sheng-Long Lv¹, Zhi-Hong Deng^{1,2}, Hang Yu¹, Ning Gao¹, and Jia-Jian Jiang¹

¹ Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

²The State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

sllv@pku.edu.cn, zhdeng@cis.pku.edu.cn,
{yh_030603, ninggao, jjj}@pku.edu.cn

Abstract. Relevance feedback algorithm is proposed to be an effective way to improve the precision of information retrieval. However, most researches about relevance feedback are based on vector space model, which can't be used in other more complicated and powerful models, such as language model and logic model. Meanwhile, other researches are conceptually restricted to the view of a query as a set of terms, and so cannot be naturally applied to more general case when the query is considered as a sequence of terms and the frequency information of a query term is considered. In this paper, we mainly focus on relevant feedback Algorithm based on language model. We use a mixture model to describe the process of generating document and use EM to solve model's parameters. Our research also employs semi-supervised learning to calculate collection model and proposes an effective way to obtain feedback from irrelevant documents to improve our algorithm.

Keywords: Relevance Feedback, Language model, Semi-supervised Learning, Irrelevant Document.

1 Introduction

Relevance feedback[9] is an interactive technique allowing users to evaluate whether a subset of documents, which usually are just returned by IR system, are relevant or not. Then the evaluation results are used to retrieve new documents. Relevance feedback is considered to be an effective way to improve performance of IR systems and has attracted much attention for a long time.

Unfortunately, most researches on relevance feedback were based on early introduced search models, such as vector space model and probabilistic model. Few work focused on implementing relevance feedback on relatively newly introduced models, for example, language model.

The language model was first introduced by Ponte and Croft. It leverages statistical methods and outperforms other search models in text retrieval area, making it an attractive text retrieval methodology. Although language modeling approach performs

well in text retrieval task, its performance on feedback is not fully explored. In most existing work, relevance feedback was implemented in the same way as vector space model by simply adding extra terms to query. This method still views query as a set of keyword rather than a statistical model, which is more compatible with language modeling approach. In the work of Zhai and J. Lafferty, a model based approach to feedback was proposed. But it only incorporated labeled relevant documents and ignored other valuable feedbacks.

In this paper, we propose a language model based relevance feedback method on KL-divergence framework. Our method gets feedback from labeled relevant documents, labeled irrelevant documents and unlabeled documents. Thus, we conclude our contribution as following:

1. We propose an detailed method to implement relevance feedback on language model.
2. Using semi-supervised learning, our method incorporates unlabeled data to estimate model parameters.
3. We explore how to incorporate labeled irrelevant documents and propose an effective solution.

In this paper, relevance feedback is executed in the following flow:

1. Given a document collection and an original query, our IR system first implements KL-divergence without feedback and returns the most relevant document according to the query.
2. IR system gets feedback that whether last returned document is relevant
3. Incorporating feedbacks, IR system implements our method and returns the most relevant document which has not been returned.
4. Go to step 2 until all the documents is returned.

This flow is the same as the flow of Relevance Feedback track of INEX 2010, which simulates use case of an internet search engine. A user inputs his query and the search engine returns a list of results. The user scans results in descending order so he first views the most relevant result the list. The user evaluates its relevance and gives feedback to search engine. Then the search engine performs another search using feedback.

This paper is organized as follows: we introduce some related work in Section 2. We present our detailed feedback model in Section 3. In Section 4, we discuss the experiment results. Finally we conclude in Section 5.

2 Related Work

2.1 Language Modeling Approach

Language model[3] is first introduced by Ponte and Croft. It is often used in natural language processing and information retrieval. Language modeling approach builds a probabilistic language model θ_d for each document. Documents are ranked based on $P(Q|\theta_d)$, the probability of its model generating the query. Since the probabilistic language model is often smoothed, making it a fine-grained model, language modeling approach often outperforms other models[16].

In [4, 10], a probabilistic model θ_q is built for query Q. Documents are ranked by the similarity of document's language model and the query model, instead of the probability $P(Q|\theta_d)$. This extension makes language model better associated with relevance feedback because we can use feedbacks to update query model, under the same idea with query expansion. In this approach, KL divergence is used to measure the similarity of two models, as showed in the following:

$$D(Q||d) = \sum_w P(w|\theta_Q) * \log \frac{P(w|\theta_Q)}{P(w|\theta_d)} \quad (1)$$

Where Q is a query and d is a document. θ_Q and θ_d are models we estimate.

Relative entropy is a non-negative value and the smaller $D(Q||d)$ is, the closer the two models are. When two models are all the same, their relative entropy is zero. Note that Relative entropy is not a legal distance function because it doesn't satisfy symmetry and triangle inequality.

2.2 Relevance Feedback Methods on Language Modeling Approach

Several papers[3,10,12,13,14,15] have focused on improving performance of language modeling approach by relevance feedback methods. In [3], the authors expanded query under the same way with Rocchio Algorithm. Terms appearing frequently in relevant documents but are relatively few in the whole collection were added to queries as a new keyword. Unfortunately, this method is limited by still viewing query as a set of keyword like VSM approach. It didn't perform all the potential of language model.

[10] and [12] focused on model-based relevance feedback methods. [10] proposed two schemes for building the query model based on a set of labeled relevant documents. The two schemes were based on regularized maximum likelihood criterion and minimizing the average KL-divergence between the query model and the relevance model. In [12], Victor Lavrenko. et al introduced a new method to construct a relevance model. It leveraged statistical methods to estimate $P(\text{tlRelevance})$ according to the terms' co-occurrence in documents. Both the two approaches improved performance of language modeling approach in their experiments. However, they only incorporated positive feedbacks but ignored labeled irrelevant documents and unlabeled documents.

There have been some works extending language modeling approach to obtain better results. Terms were usually considered independent in existing work. Dependence models extends language model by taking terms' relationship into consideration[15]. A cluster-based term selection algorithm was proposed in [14] for constructing refined query language model. [13] presented a learning approach to balance the original query and feedback information while most methods set this balance parameter to a fixed value.

3 Model Definition

In our approach, the fundamental task is to estimate query model and each document's model as most KL-divergence approach. The query model should involve

valuable feedback from labeled documents and global information from the whole collection.

3.1 Incorporating Labeled Relevant Documents

Intuitively, most relevant documents belong to the same topic. Thus, we assume there is a relevance model generates all the relevant document. To incorporate labeled relevant document, a generative relevance model θ_R should be estimated using feedbacks. Then our new query model is:

$$\theta'_R = \theta_Q + \theta_R \tag{2}$$

Where θ_Q is the original query model and θ_R is the relevance model.

Now our challenge is how to estimate relevance model θ_R . We assume each relevant document is a sample of the relevance model, then a natural way to estimate relevance model is to maximize the probability that relevance model generating all the relevant documents. The probability is:

$$P(D_r|\theta_R) = \prod_{d_i \in D_r} \prod_w P(w|\theta_R)^{c(w,d_i)} \tag{3}$$

Where D_r is the set of labeled relevant documents. $c(w, d_i)$ denotes the number of term w that appears in document d_i .

The equation 3 is sufficient if there is only relevant information in relevant documents. However, relevant documents often contain trivial parts. These parts neither do harm to nor contribute to the relevance of document. As a consequence, a more reasonable model would be a mixture model of relevance model and collection model. The collection model generates all the trivial part of the collection. This means that a term w can be generated by relevance model by probability $P(w|\theta_R)$, or can be generated by collection model by probability $P(w|\theta_C)$. We also assume a term in a document can only be generated by one model. So this problem can be viewed as a classification problem. Terms are classified into two classes: relevance and collection. We use $I(\theta, w)$ to denotes our classification result. It denotes the probability that term w is generated by model θ . Under this assumption the log-likelihood of labeled relevant documents is:

$$\log L(\theta) = \sum_{d_i \in D_r} \sum_w c(w, d_i) (P_{w,R} + P_{w,C}) \tag{4}$$

$$P_{w,R} = I(\theta_R, w) \log \tau_r P(w|\theta_R) \tag{5}$$

$$P_{w,C} = I(\theta_C, w) \log \tau_c P(w|\theta_C) \tag{6}$$

Where τ_r denotes the probability that relevance model θ_R generates a term, and τ_c denotes the probability that collection model θ_C generates a term.

$I(\theta, w)$ and τ are maybe difficult to distinguish. $I(\theta, w)$ is our classification result as a hidden variable in the model and τ is model parameter which denotes the intrinsic feature of the model. And for each term $I(\theta, w)$ is different but for all the term τ is all the same.

How to maximize equation 4 will be presented in the next section.

3.2 Incorporating Unlabeled Documents

The basic model did not give a reasonable definition and interpretation of collection model θ_C . Under our definition of collection model, we should use all the trivial part of the collection to estimate collection model. However, it is difficult to identify which part is trivial in a document. Another way to solve this problem is to use all the unlabeled documents instead of trivial parts. This is because most parts of an irrelevant document are trivial parts and most of unlabeled documents are irrelevant. Thus, we should maximize the probability that collection model generates unlabeled document:

$$P(D_u|\theta_C) = \prod_{d_i \in D_r} \prod_w P(w|\theta_C)^{c(w,d_i)} \quad (7)$$

Where D_u is the set of labeled relevant documents.

Thus, the new likelihood function is

$$\log L(\theta) = \sum_{d_i \in D_r} \sum_w c(w, d_i) (P_{w,R} + P_{w,C}) + \sum_{d_j \in D_u} \sum_w c(w, d_j) P_{w,C} \quad (8)$$

$$P_{w,C} = \log \tau_C P(w|\theta_C) \quad (9)$$

The newly added part in equation 8 is the log-likelihood of unlabeled documents. In the calculation of this part, we believe that all of the terms are generated by the collection model, so this is not a classification problem and $I(\theta, w)$ isn't used as a factor here.

We use EM (Expectation-Maximization) Algorithm[8] to maximize likelihood function and solve the parameter. Now the model parameters are $\tau_r, \tau_c, P(w|\theta_R)$ and $P(w|\theta_C)$ and the hidden parameters are $I(\theta_R, w)$ and $I(\theta_C, w)$. EM is an iterative algorithm. Each iteration consists of two steps: In E step hidden variables are estimated by the current model parameters' values; In M steps model parameters are re-estimated according the hidden values to maximize the likelihood function. After each iteration, the likelihood function is larger than the prior iteration. Meanwhile, as the gradual process of maximizing, EM algorithm will find a good local optimal solution in most cases.

In E step, hidden variable $I(\theta, w)$ is:

$$I(\theta_R, w) = \frac{\tau_r P(w|\theta_R)}{\tau_r P(w|\theta_R) + \tau_c P(w|\theta_C)} \quad (10)$$

$$I(\theta_C, w) = 1 - I(\theta_R, w) \quad (11)$$

In M step, using the value of $I(\theta_R, w)$ and $I(\theta_C, w)$ obtained in E step, we make the likelihood function maximum and estimate model parameters' values by employing Lagrange constrained optimization method. Here we have three constraints:

$$\tau_r + \tau_c = 0 \quad (12)$$

$$\sum_w P(w|\theta_R) = 1 \quad (13)$$

$$\sum_w P(w|\theta_C) = 1 \quad (14)$$

Considering these three constrains, we can get the optimal value of model parameters. The results are:

$$\tau_r = \frac{\sum_{d_i \in D_r} \sum_w c(w, d_i) I(\theta_R, w)}{\sum_{d_i \in D_r} \sum_w c(w, d_i) I(\theta_R, w) + \sum_{d_i \in D_r} \sum_w c(w, d_i) I(\theta_C, w) + \sum_{d_j \in D_u} \sum_w c(w, d_j) I(\theta_C, w)} \quad (15)$$

$$\tau_c = 1 - \tau_r \quad (16)$$

$$P(w|\theta_R) = \frac{\sum_{d_i \in D_r} c(w, d_i) I(\theta_R, w)}{\sum_w \sum_{d_i \in D_r} c(w, d_i) I(\theta_R, w)} \quad (17)$$

$$P(w|\theta_C) = \frac{\sum_{d_i \in D_r} c(w, d_i) I(\theta_C, w) + \sum_{d_j \in D_u} c(w, d_j) I(\theta_C, w)}{\sum_w (\sum_{d_i \in D_r} c(w, d_i) I(\theta_C, w) + \sum_{d_j \in D_u} c(w, d_j) I(\theta_C, w))} \quad (18)$$

In this paper, the termination condition is:

$$\sum_w (\Delta P(w|\theta_R) + \Delta P(w|\theta_C)) < 10^{-4} \quad (19)$$

During calculation, we encountered the same problem as [10]. If we estimate these four parameters at the same time, τ_c will be very close to zero. Then our model degenerated to a unigram model. However, this problem didn't have bad impact on the final results. But we would like to have a non-zero τ_c to make our model more reasonable. So we set τ_c and τ_r to same constant during calculation.

3.3 Incorporating Labeled Irrelevant Documents

In our model, last returned document is always ranks first in all the documents which have not been returned. Then it is a more valuable feedback that if last returned document is labeled irrelevant, because last returned document is the most relevant document according to our estimation. That it is labeled irrelevant denotes our estimation can't perform properly, so our model needs modification.

Because last returned document is the most relevant document according to our estimation, so this irrelevant document must contain a number of highly relevant terms. In our model, the relevance of term is determined by the $P(w|\theta_R)$ indirectly, so this document must have some terms whose $P(w|\theta_R)$ are relatively high, making document model close to the query model. However, $P(w|\theta_R)$ is calculated from the set of labeled relevant documents and of very high credibility. Therefore, we believe that these terms whose $P(w|\theta_R)$ is high do no harm to the document's relevance. It is some of other terms that frequently appear in this document do harm to the document's relevance. And these words are of high irrelevance features so that the document that contains many relevant terms can be labeled irrelevant. In this paper, we call this kind of terms punishing terms. We extract punishing terms from irrelevant documents and punish the relevance of a document if it contains punishing terms.

For example, when a user enters the query "Nobel Prize" to find some introduction of Nobel Prize, a document about Ig Nobel Prize is labeled as irrelevant. This shows that the user wants to query the real Nobel Prize information. However, the document contains large numbers "Ig Nobel Prize" and the relevance of "Nobel" and "Prize" is very high. Traditional Relevance Feedback algorithm like Rocchio Algorithm would punish these three words at the same degree. In our model, because "Nobel" and

"Prize" is highly relevant, we mainly punish the relevance of "Ig", which is fully in line with the user's search intent. This shows the advantages of our model.

When last returned document is labeled as irrelevant, we use the following method to extract the punishing terms:

1. For each term w in the document, calculate the value of $P(w|\theta_d) - P(w|\theta_R)$.
2. Rank terms in descending order according to the value.
3. Select the top m term as punishing terms.
4. Record $P(w|\theta_R)$ of every punishing terms.

We extracted m from each irrelevant documents and build the punishing model θ_P as follows:

$$P(w|\theta_P) = \frac{\sum_{d \in D_i} P(w|\theta_d)}{\sum_w P(w|\theta_P)} \quad (20)$$

Where D_i is the set of labeled irrelevant documents.

When last returned document is labeled as irrelevant, if there are still relevant documents that have not been returned, the feedback is sufficient to denotes that our model can't recognize irrelevant features properly. When there is no relevant documents that have not been returned, returning a irrelevant document is inevitable. Extracting punishing word from that document may affect our model's effectiveness. Therefore, we choose the first n irrelevant documents to extract punishing terms.

After the punishing model has been built, we use the relative entropy to compute the irrelevance of a document:

$$D_P(d||P) = \sum_w P(w|\theta_P) * \log \frac{P(w|\theta_P)}{P(w|\theta_d)} \quad (21)$$

3.4 Rank the Documents

In [11], Robertson proposed that documents can be sorted by the odds of their being observed in relevant class and irrelevant class, namely:

$$\frac{P(D|R)}{P(D|N)} \quad (22)$$

Where, R represents the document is relevant while N represents the document is not relevant.

In this paper, we follow Robertson's sorting methods, using the odds of document relevance score and document punishing score as the sorting criteria.

$$\frac{D_r(d||Q)}{D_P(d||P)^\alpha} \quad (23)$$

Where α controls the weight of punishment.

According to previous research, experimental results are often better when irrelevant feedbacks have a smaller weight than relevant feedbacks. Thus, in our experiment, we set $\alpha = 0.5$.

4 Experiment

4.1 Dataset and Competitors

We choose dataset of Relevance Feedback Track of INEX 2009 as our experiment dataset. This dataset was downloaded from Wikipedia at October, 2008 and labeled by YOGO (2008-w40-2). The size of dataset is 50.79GB. Documents in this dataset are all in XML format. Every document's relevance has been manually evaluated. And the evaluation is accurate to sentences, so we could know which sentence is relevant in a relevant document.

In our experiment, we randomly choose six topic to run our model. Following is each topic's information:

Table 1. Fig. 1. Information of topics

<i>Topic ID</i>	<i>Query</i>	<i>Document Count</i>	<i>Relevant Count</i>	<i>Relevant Percent</i>
2009023	Plays of Shakespeare Macbeth	751	86	0.1145
2009088	hatha yoga deity asana	754	13	0.0172
2009089	world wide web history	758	71	0.0937
2009095	Weka software	753	19	0.0252
2009109	circus acts skills	753	155	0.1955
2009115	virtual museums	759	61	0.0803
Sum		4528	405	0.0894

In our experiment, we compared our model to Rocchio Algorithm[6], KL-divergence approach. And we will analyze the impact of incorporating different set of documents.

Rocchio Algorithm has been proven stable and excellent as a relevance feedback approach in many experiments. Thus comparing with Rocchio Algorithm would give strong evidence of the performance of our model. Our model is based on KL-divergence, so comparing with KL-divergence will show how much the performance improve after acquiring feedbacks.

4.2 Experiment Results

In our experiment, we treated each document as plain text by filtering all the tags in documents. And a document is considered relevant as long as it contains relevant sentences.

Our model vs KL-divergence

Our model is based on KL-divergence, so we run our model two times in this experiment. The first run has no feedbacks and the second run has feedbacks to update models. The following are experiment results:

Table 2. Results of our model vs. KL-divergence

	P@5	P@10	P@20	P@50
KL-divergence	0.57	0.50	0.51	0.35
Our Model	0.70	0.72	0.56	0.44

The table 2 shows our model significantly improves the precision of KL-divergence, especially at the low recall rate. This denotes our model is an effective method for language modeling approach to incorporate feedbacks. Note that as the number of documents increases, the improvement of precision gets smaller.

VSM vs. KL-divergence

Before comparing our model and Rocchio Algorithm, we first run VSM approach and KL-divergence to obtain their performance. This is because our model and Rocchio Algorithm are based on different models. Our model is based on KL-divergence approach while Rocchio Algorithm is based on VSM approach. If there is a big difference between the effect of VSM approach and KL-divergence, our model and Rocchio Algorithm are not comparable. Comparison results are as follows:

Table 3. Results of VSM vs. KL-divergence

	P@5	P@10	P@20	P@50
VSM	0.51	0.52	0.39	0.33
KL-divergence	0.57	0.50	0.51	0.35

As can be seen from the table, KL-divergence is slightly better than VSM approach. This is because KL-divergence approach is based on statistical methods and relaxes restrictions of relevant documents on the query model, making it more suitable to apply to general case.

Our model vs. Rocchio Algorithm

In our experiments, we implemented Rocchio Algorithm as the following equation[7]:

$$Q' = \alpha Q_0 + \frac{\beta}{n_r} \sum_{d \in D_r} v_d - \frac{\gamma}{n_s} \sum_{d \in D_s} v_d \quad (24)$$

Our Rocchio Algorithm got feedbacks from both labeled relevant document and labeled irrelevant document. According to previous experiment, we set $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.5$, Table 3 shows the experiment results.

Table 4. Results of our model vs. Rocchio

	P@5	P@10	P@20	P@50
Our Model	0.70	0.72	0.56	0.44
Rocchio	0.59	0.60	0.59	0.58

Table 4 shows that before 20 feedback documents our model is significantly better than Rocchio algorithm. At 5 feedback documents, our model improves KL-divergece by 23% while Rocchio algorithm improves VSM approach by 15%. And at this point, KL-divergence is better than VSM approach by 12%. We know that the better the results of a rmodel are, the more difficult to improve it . At 10 feedback documents, our model improves KL-divergece by 44% while Rocchio algorithm improves VSM approach by 15%.

However, at 50 feedback documents, our model isn't better than Rocchio Algorithm any longer. The reason for the decline is likely to be that our model gets a poor local optimal solution during the process of maximizing the expectation. This is because Rocchio algorithm uses the vector to update the query vector, that is, each of the relevant documents has the same weight to change query vector. Our model uses EM method iteratively update query model. After getting a large number of feedback documents, newly returned document is chosen according to similarity with query model, thus its document model changes query model little then, so it is easy to fall into local optimal solution.

5 Conclusion and Future Work

In this paper, we propose a relevance feedback model based on language modeling approach. Our model fully utilizes feedbacks including labeled relevant documents, labeled irrelevant documents and unlabeled documents to estimate a new query model. We employ EM Algorithm to solve model parameters, semi-supervised learning method for modeling the document collection and build punishing model to get valuable information from labeled irrelevant feedback documents.

Experiments show our model improves the precision of language modeling approach. Especially at the lower recall rate, the improvement is significant. However, at higher recall rate our improvement drops and finally the precision is lower than Rocchio Algorithm. But our model is still an effective approach for KL-divergence to incorporate feedbacks. Because users tend to access only the first few pages of search results when using internet search engine. The recall of first few pages is very low.

Acknowledgement. This work is partially supported by Project 61170091 supported by National Natural Science Foundation of China and Project 2009AA01Z136 supported by the National High Technology Research and Development Program of China (863 Program).

References

1. van Rijsbergen, C.J.: Information retrieval, 2nd edn. Butterworths (1979)
2. Crestani, F., Ruthven, I., Sanderson, M., van Rijsbergen, C.J.: The troubles with using a logical model of IR on a large collection of documents. In: Harman, D.K. (ed.) Proceedings of the Fourth Text Retrieval Conference (TREC-4), pp. 509–525. NIST special publication (1995)

3. Ponte, J., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the ACM SIGIR 1988, pp. 275–281 (1988)
4. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of SIGIR 2001 (2001)
5. Lafferty, J., Zhai, C.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR 2001 (2001)
6. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System Experiments in Automatic Document Processing, ch. 14, pp. 313–323 (1971)
7. Ide, E., Salton, G.: Interactive search strategies and dynamic file organization in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System - Experiments in Automatic Document Processing, ch. 18, pp. 373–393 (1971)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statist. Soc. B* 39, 1–38 (1977)
9. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41(4), 288–297 (1990)
10. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of CIKM 2001 (2001)
11. Robertson, S.E.: *The Probability Ranking Principle in IR*, pp. 281–286. Morgan Kaufmann Publishers, Inc., San Francisco (1997)
12. Lavrenko, V., Croft, B.: Relevance-based language models. In: Proceedings of SIGIR 2001 (2001)
13. Lv, Y., Zhai, C.: Adaptive relevance feedback in information retrieval. In: Proceedings of CIKM 2009 (2009)
14. Tan, B., Velivelli, A., Fang, H., Zhai, C.: Term feedback for information retrieval with language models. In: Proceedings of SIGIR 2007 (2007)
15. Bai, J., Song, D., Bruza, P., Nie, J.-Y., Cao, G.: Query expansion using term relationships in language models for information retrieval. In: Proceedings of CIKM 2005 (2005)
16. Bennett, G., Scholer, F., Uitdenbogerd, A.: A comparative study of probabilistic and language models for information retrieval. In: Nineteenth Australasian Database Conference (ADC). CRPIT, ACS, vol. 75, pp. 65–74 (2008)
17. Xu, Z., Akella, R.: A bayesian logistic regression model for active relevance feedback. In: Proceedings of SIGIR 2008 (2008)

FXProj – A Fuzzy XML Documents Projected Clustering Based on Structure and Content

Tengfei Ji, Xiaoyuan Bao, and Dongqing Yang

Peking University
Beijing, China
{tfji,xybao,dqyang}@pku.edu.cn

Abstract. XML documents possess inherent semi-structured property, consisting of structural and content features. Most existing methods for XML documents clustering consider only one aspect of them. In this paper, we propose a fuzzy XML documents projected clustering algorithm, which can be used to cluster XML documents efficiently by combining the structural and content features. Another contribution is the adoption of some fuzzy techniques in a way that each frequent induced substructure has a fuzzy parameter associated with each cluster. Experimental results on both synthetic and real datasets show its effectiveness, especially when applying to large schemaless XML document collections.

1 Introduction

Due to the inherent semi-structured nature, XML (eXtensible Markup Language) has emerged as a standard for information representation and exchange on the Web. Recently, the wide use of the web has speeded up the research on management and analysis of XML data. Therefore, further to storing and querying of XML documents, mining of XML documents has become a new trend. XML clustering is a task of grouping similar XML data across heterogeneous ones without prior knowledge [1]. The clustering of XML documents is useful in information retrieval, database indexing, data integration and document engineering [2].

Compared with text mining, XML document clustering is a challenging task because it involves content as well as structure information. There are some methods for XML documents use either the structural features [3, 4] or the content features [5] for clustering similar documents. Some literatures have proved that the performance of grouping XML documents only by their content features couldn't satisfy actual application. Sometimes, most of documents are produced by only a few schemas. On this and like occasions, grouping XML documents based only on their structural features most probably leads to incorrect results. A simple example in Figure 1 will serve to illustrate this point. Obviously, the XML documents in Figure 1. can be divided into three categories depending on the structural and content features: (a) (c), (b) (d) and (e) (f). However, the content-only methods will group (b) and (c) together incorrectly, since no attention is paid to the structure. It can be noted that though (a) (c) and (b)

<pre><Book> <Title>Data mining approaches</Title> <Author>Krzysztof J.Cios</Author> <Author>Witold Pedrycz</Author> <Publisher>Springer</Publisher> </Book></pre>	<pre><Forum> <Topic>Recent Advances in Data Mining</Topic> <Speaker>Max Bramer</Speaker> <Sponsor>Springer</Sponsor> </Forum></pre>
(a)	(b)
<pre><Book> <Title>Principles of Data Mining</Title> <Author>Max Bramer</Author> <Publisher>Springer</Publisher> <ISBN>0387333339</ISBN> </Book></pre>	<pre><Forum> <Topic>Advanced Data Mining and Applications</Topic> <Speaker>Jianyong Wang</Speaker> <Sponsor>Tsinghua University</Sponsor> </Forum></pre>
(c)	(d)
<pre><Book> <Title>Principles of Plymer Chemistry</Title> <Author>Paul J.Flory</Author> <Publisher>Cornell Univ Pr</Publisher> <Year>1953</Year> </Book></pre>	<pre><Book> <Title>Contemporary Plymer Chemistry</Title> <Author>Harry Allcock</Author> <Author>Fred Lampe</Author> <Author>James Mark</Author> <Publisher>Prentice Hall</Publisher> <Year>2003</Year> </Book></pre>
(e)	(f)

Fig. 1. The fragments of six XML documents

(f) share a similar structure, they are different in content, which is neglected by the structure-only methods.

To correctly identify similarity among documents, the clustering process should use both their structure and their content information. Approaches on clustering both the structure and the content features of the XML documents are limited [5].

The proposed fuzzy XML documents projected clustering algorithm (FXProj) not only aims to combine the structure and content of XML documents efficiently, but also to take advantage of some fuzzy clustering technology that helps to avoid the difficulty of choosing appropriate frequent induced substructures, resembling projected dimensions in projected subspace, for each cluster during the iterations. In FXproj, each frequent induced substructure has a weight associated with each cluster indicating the degree of importance to the cluster. Several experiments are conducted to show the proposed algorithm has a good performance, especially when applying to large schemaless XML document collections.

The rest part of this work is organized as follows: Section 2 discusses the recent related work; Section 3 gives the preliminaries about XProj algorithm and LAC algorithm; Section 4 proposes our fuzzy XML documents projected clustering based on structure and content; Section 5 gives experiments for our approach on both real and synthetic data sets, and shows the achieved results. Section 6 makes a conclusion about the whole work.

2 Related Works

There has been a myriad of clustering algorithms for XML documents proposed in recent years, which may be arranged under three heads.

XML Documents Clustering Based on Content Features: The existing approaches put forward three ways to group XML documents by using content features. (1) Embedding some special query language, e.g. XQuery, in application programs. However, it brings a lot of complexities and enormous expenses. (2) Mapping XML documents to relational data models. The main disadvantage is ignoring the inherent semi-structure information of XML, which is capable of causing infractions during mapping process. (3) Considering XML documents as texts then clustering them by traditional text mining techniques. This method fails to take account of semi-structure information of XML either.

XML Documents Clustering Based on Structure Features: In this area, literatures are principally concerned with two aspects. (1) The representation of XML documents. The layout of the document can vary and could be shown as respectively a tree, a graph, sets of paths, time series, vectors and others. Most of the existing models for the representation of XML documents are based on labeled tree, as it is a natural illustration indicating a hierarchical structure of XML documents [7]. (2) The similarity measures and clustering by structure. The earliest work on clustering tree structured data was designed to cluster XML schemas [1]. However, it is known that only 48% of documents contain links to specific schemas [8]. Consequently, integrating the enormous volume of schemaless and semantically different documents to realize a Web database is a breath-taking task [9]. If the solution is based on the tree, the authors have used tree edit distance to measure the similarity between the structures of documents [7]. Joe Tekli et al provides a survey about similarity measures for XML documents in [10].

XML Documents Clustering Based on both Content Features and Structure Features: Despite the advantages, there are only a few solutions proposed using both structural and content features, since how to combine structural and content features effectively for scalable clustering is still a serious challenge. The typical approaches of this kind are XCFS [2], HCX [11] and SCVM [12].

3 Preliminaries

3.1 XProj Algorithm

Since the structural aspects of the XML documents result in a high implicit dimensionality of the data representation, XProj algorithm [1] employs a projection based structural approach. It adopts frequent substructures of the underlying documents to measure the similarity, which is analogous to the concept of projected clustering in multi-dimensional data. In addition, choosing the rank of the substructures is analogous to choosing the dimensionality of the projection in the case of projected clustering.

The structural similarity $\Delta(R, T)$ of the set of documents $R = \{R_1, R_2, \dots, R_j\}$ to the set of frequent structures $T = \{T_1, T_2, \dots, T_k\}$ is defined as $\Delta(R, T) = \frac{\sum_{i=1}^j \delta(R_i, T)}{j}$, where $\Delta(R, T)$ is defined to be the fraction of nodes in R which are covered by some structure in T.

XProj uses a set of frequent substructures as the representative rather than a single representative, which strengthens its robustness. To speed up the frequent substructure representative mining, XProj adopts a set of high quality approximate structures, that is, sequences of tree edges. The selection of representative substructures is based on the sequential covering paradigm.

3.2 LAC Algorithm

LAC [13] defines subspace clusters as weighted clusters such that each cluster consists of a subset of data points together with a vector of weights. That means the concept of cluster is not based only on a data set of n points D , but also involves a weighted distance. To be precise, LAC defines the j th ($1 \leq j \leq k$) cluster as $C_j = \{x \in D : (\sum_{i=1}^d w_{ji}(x_i - z_{ji})^2)^{\frac{1}{2}} < (\sum_{i=1}^d w_{li}(x_i - z_{li})^2)^{\frac{1}{2}}, \forall l \neq j\}$, where z_{ji} is the center and w_{ji} is the component of weight vector. The centers and weights are chose such that the error measure, $E = \sum_{j=1}^k \sum_{i=1}^d w_{ji}e^{X_{ji}}$, is minimized

$$\sum_{i=1}^d w_{ji}^2 = 1, \forall j, \text{ where } X_{ji} = \frac{1}{|C_j|} \sum_{x \in C_j} (x_i - z_{ji})^2.$$

4 The FXProj Algorithm

Our FXProj clustering algorithm falls into three major phases, which is illustrated in Figure 2. In the first phase, XML documents are modeled as document trees, which display the structure information of XML documents. The second phase is content mining. FXProj extracts the content according to frequent induced substructures generated in the first phase. The last phase of FXProj is clustering XML documents based on both structure features and content features.

4.1 Phase I: Structure Mining

To display the hierarchical relationships between nodes in the document, we model a set of XML documents $D = \{D_1, D_2, \dots, D_n\}$ as a corresponding set of ordered, labeled and rooted document trees $DT = \{DT_1, DT_2, \dots, DT_n\}$. We do not distinguish between attributes and elements of an XML document, since both are mapped to the label set.

DEFINITION 1: Frequent Induced Substructure

A frequent induced substructure T of an XML document tree DT, is an undirected, connected, labeled, acyclic graph that satisfies the following conditions:

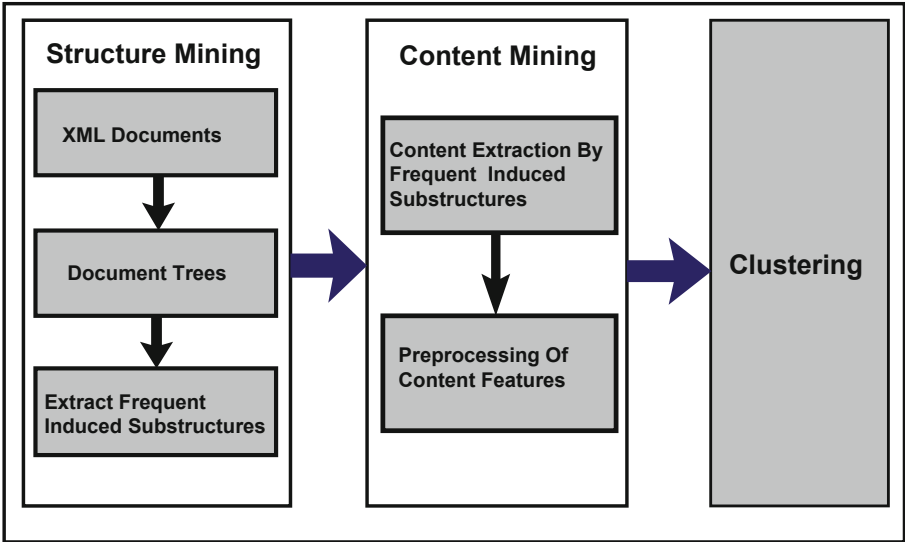


Fig. 2. The phases of FXProj algorithm

(1) The vertices and edges in T can be one-to-one mapped to a subset of vertices and edges of DT , (2) and it preserves the vertex labels and ancestor-descendant relationships among the corresponding vertices, (3) the occurrence of substructure is up to the user defined minimum support.

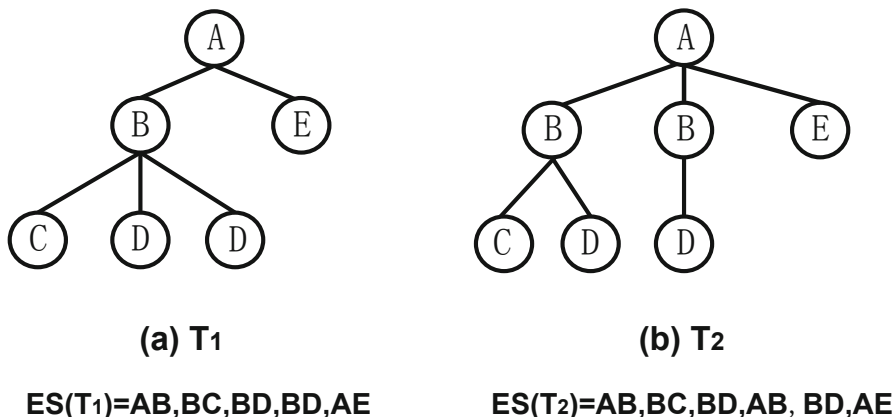
Since it has been proved in XProj algorithm that using *sets of substructures* S as representatives rather than individual XML documents is capable of enhancing robustness, our FXProj still adopts this idea here. The set of frequent induced substructures of size l are regarded as the representatives for partitions.

DEFINITION 2: Coverage of Frequent Induced Substructure

A node x in the document tree DT is said to be covered by a set of frequent induced substructures T of size l , $T = \{T_1, \dots, T_k\}$, if there exists correspondence from each node in $T_i \in T$ to a node in DT .

Due to the graph isomorphism problem, determining the frequent substructures of size l as representatives is a main time-consuming operation. Instead of exact approach, the approximate data representation could be adopted to avoid the graph isomorphism problem. Compared to node sequence representation, edge sequence representation that obtained by preorder depth-first traversal maintains more structural information. For this reason, XProj introduced edge sequence representation produced by preorder depth-first traversal, which, unfortunately, may cause the *false mapping problem* [1]. Figure 3 illustrates the false mapping problem. ES is the abbreviation for edge sequence.

To solve above mentioned false mapping problem, we develop a new edge sequence representation by adding the number of children to the starting point of each edge. For example, the edge sequence of T_1 in Figure 2, $ES(T_1)$, is



False Mapping: $T_1 \not\subset T_2$ but $ES(T_1) \subset ES(T_2)$

Fig. 3. An example of false mapping

AB_2,BC_3,BD_3,BD_3,AE_2 . The first figure "2" means node A has two children, namely B and E. Similarly, the second figure "3" denotes that node B has three children C, D and D. Analogously, $ES(T_2)$ is $AB_3,BC_2, BD_2,BD_1,AE_3$.

The property of edge sequence representation can be expressed as follows: A tree T_1 is a subtree of another tree T_2 iff (1) the edge sequence representation of T_1 must be a subsequence of the edge sequence representation of T_2 , (2) each number that follows each edge in T_1 must be smaller than the number that follows the corresponding edge in T_2 . The example in Figure 3 is a false mapping because it breaks the second condition. In $ES(T_1)$, the numbers follow the edges BC,BD,BD are 3,3,3 respectively, which are larger than those in $ES(T_2)$ 2,2,1. By adding the number of children to the starting point of each edge, our new edge sequence representation removes the false mapping problem and reduces potential substructure relationship. Obviously, if a substructure is frequent, its corresponding edge sequence must be frequent too.

From the above analysis, we can see that frequent induced substructures need to cover as many nodes as possible, which is analogous to the significance of projected space in high-dimensional projected clustering. In high-dimensional projected clustering, several clusters may exist in different subspaces comprised of different combinations of dimensions. Each dimension could be relevant to at least one of the clusters, and some approaches have proved that fuzzy technique is an answer to this difficulty. In fact, such difficulty also exists in XML documents clustering, that is, the frequent induced substructures may exist in different clusters. Therefore, we introduce some fuzzy technique here to solve this problem.

DEFINITION 3: Structural Distance

The structural distance $\delta(R_i, T)$ of an XML document $R_i \in R$ to a set of frequent induced substructures $T = \{T_1, \dots, T_k\}$ is defined to be the section of nodes in R_i

that are uncovered by any T_j ($1 \leq j \leq k$) in T . Let us assume the representative of the j th cluster is T_j , which contains d frequent induced substructures, $T_j = \{T_{j1}, \dots, T_{jd}\}$.

DEFINITION 4: Objective of Structure Clustering

The objective of structure clustering is defined as to minimize the following function F .

$$F = \frac{\sum_{j=1}^k \sum_{i=1}^n \sum_{l=1}^d w_{jl}^\alpha \delta(R_i, T_{jl})}{n} \tag{1}$$

where the l th frequent induced substructure is associated with the j th cluster to a degree of w_{jl} . $\alpha \in (1, \infty)$ is a fuzzier. The fuzzy-coefficient w_{jl} is an item of $k \times d$ fuzzy-coefficient matrix W , which satisfies the following conditions:

$$\sum_{l=1}^d w_{jl} = 1 \quad 1 \leq j \leq k$$

$$0 \leq w_{jl} \leq 1 \quad 1 \leq j \leq k, 1 \leq l \leq d$$

The above definition illustrates that FXProj assigns fuzzy-coefficients according to the coverage of frequent induced substructure. Frequent induced substructures along which cluster are loosely correlated receive a small fuzzy-coefficient. On the contrary, those frequent induced substructures which own extensive coverage receive a large fuzzy-coefficient.

To find the fuzzy-coefficient matrix W given the estimate of T such that the function F is minimized, we use the method of Lagrange multipliers. To do this, we first write the Lagrangian function as

$$\Omega(W, \Lambda) = F - \sum_{j=1}^k \lambda_j \left(\sum_{l=1}^d w_{jl} - 1 \right)$$

By taking partial derivatives, we obtain

$$\frac{\partial \Omega(W, \Lambda)}{\partial w_{jl}} = \frac{\alpha}{n} w_{jl}^{\alpha-1} \delta(R_i, T_{jl}) - \lambda_j = 0 \quad 1 \leq j \leq k, 1 \leq l \leq d$$

$$\frac{\partial \Omega(W, \Lambda)}{\partial \lambda_j} = \sum_{l=1}^d w_{jl} - 1 = 0 \quad 1 \leq j \leq k$$

which, with some simple manipulations, leads to

$$w_{jl} = \frac{\delta(R_i, T_{jl})^{\alpha-1}}{\sum_{j=1}^k \delta(R_i, T_{jl})^{\alpha-1}} \quad 1 \leq j \leq k, 1 \leq l \leq d \tag{2}$$

4.2 Phase II: Content Mining

The first task of this phase is to eliminate the content corresponding to infrequent substructures such that the dimensionality of the input (or content) data is reduced. Only those terms whose nodes are covered by frequent induced substructures are extracted. As structure information of XML documents, the frequent induced substructures obtained from phase I are utilized to constrain the content of XML documents. Hence, FXProj offers a nice combination of structure features and content features of XML documents.

Before implementing XML document clustering, the extracted terms (or content) are expected to be preprocessed. We also adopt several pre-processing methods such as stop word removal and stemming which are most widely used in IR technique [14]. Here we just provide a brief introduction to these preprocessing methods.

Stop word Removal: High frequency words with low information content are called stop words, such as "the", "a" and "of". Because of weak discriminating power, these stop words should be removed before conducting clustering.

Stemming: Word stemming is a process reducing morphological variants to the root word. For example, the word "moves", "moving" and "moved" are replaced by "move", which could reduce the number of distinctive words.

Then we adopt *tf*idf* method (term frequency * inverse document frequency) to assign the weights of terms, which is a popular method introduced by [15]. The main principle of this method is: The weight of a term t in XML document D_i ($D_i \in D$) is directly proportional to the number of its occurrence in D_i and inversely proportional to its occurrence in the entire XML documents collection D . In other words, if a term t occurs frequently in a certain XML document D_i , but it occurs rarely in other XML documents, this means the term t is such a distinctive one that deserves a high weight. For more details of the *tf * idf* method, we refer to [15].

4.3 Phase III: Clustering

The quintuple (Fuzzy-coefficient, Frequent induced substructure, Term weight, Term, Document) generated from the previous phases becomes input to a partition clustering algorithm. Term in the quintuple represents the terms extracted corresponding to the frequent induced substructures. Since the amount of frequent induced substructure is far less than that of the entire collection, the efficiency of our clustering algorithm is enhanced. Fuzzy-coefficient in the quintuple relaxes the restriction that the association between a frequent induced substructure and a cluster is either 0 or 1. The quintuple that consists of the structure features and the content features of XML documents is used in this phase to compute the similarity between the XML documents for the clustering.

Algorithm: FXProj Algorithm (High level definition)

Input: Document Tree Set: DT, Minimum support min_sup , Constraint Length: l, The Number of Clusters: K;

Output: Clusters: $\{C_1, C_2, \dots, C_K\}$

Step 1: Initialize representative sets S: $\{S_1, S_2, \dots, S_K\}$ and Fuzzy coefficient W;

Step 2: Scan DT and assign each document tree to S_i ($1 \leq i \leq K$).

Step 3: Find all l-length frequent induced substructures for the given min_sup using coverage based similarity criterion;

Step 4: Assign fuzzy coefficient for frequent induced substructure according to Equation (2);

Step 5: Extract the content information Term according to l-length frequent induced substructure;

Step 6: Compute weights of terms using tf-idf method;

Step 7: Apply K-means clustering to quintuple (Fuzzy-coefficient, Frequent induced substructure, Term weight, Term, Document) to generate the K number of clusters.

5 Experiments

In this section, we illustrate the general behavior of the proposed FXProj algorithm. We evaluate our algorithm by using a PC with 2.2 GHz Pentium(R) Dual-Core CPU and 2G of memory, running Win7, and programmed by Java.

5.1 Data Description and Evaluation Measures

We conducted experiments on both real and synthetic data sets to compare our algorithm with other algorithms.

Real Data Sets: The ACM SIGMOD is a small dataset that contains 140 XML documents corresponding to 2 DTDs. When only structure similarity is taken into account, ACM SIGMOD dataset will be divided into two groups. However, considering both the structure and the content similarity, the dataset ought to have five categories using expert knowledge [6]. The second real data set is the INEX 2007 Wikipedia XML Corpus, which is a subset of Wikipedia. Unlike the ACM SIGMOD dataset, the INEX 2007 dataset is composed of more than 600,000 XML documents without any schemas. The dataset is supposed to be divided into 21 groups considering both structure features and content features.

Synthetic Data Sets: We used the XML generator that introduced in [16] to generate three synthetic datasets. To be fair, each DTD was used to generate 100 documents. More detail information about these synthetic datasets is shown in Table 1.

It can be noted that SYN1 dataset on the above table was used to evaluate clustering performance only by the structure features, namely the number of DTDs. On the contrary, SYN2 and SYN3 datasets were generated to assess clustering performance by both the structure features and the content features. Therefore, cluster and DTD are not equal in number.

Table 1. Summary of the Synthetic Datasets

Dataset	Size	Cluster	DTD
SYN1	300	3	3
SYN2	300	7	3
SYN3	1,000	20	10

We measured the performance of different algorithms using well-known metric F1 measure, which is defined as follows.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

where recall is ratio between the number of correct positive predictions and the number of positive examples; precision is ratio between the number of correct positive predictions and the number of positive predictions.

5.2 The Accuracy of FXProj Algorithm

To evaluate the clustering performance, we compared FXProj algorithm against three other XML clustering algorithms. The first approach only considers the structure features by using SOM (Self-Organizing Map), which is denoted as the Structure Approach (SA). The second one is a traditional content-based clustering method VSM, which uses vector space model and tf-idf weight. In addition, we compared our algorithm with XProj, which is also a structure-only approach. We compared the F1 of the four algorithms on both real and simulated data sets.

In fairness to all algorithms, each algorithm was supposed to run 20 times on each data set. The fuzzy parameter α in FXProj algorithm was set to be 1.8. The experiments were conducted with *min_sup* at 0.2 and frequent induced substructure length at 4. Table 2 and Table 3 illustrate the comparison results on real datasets and synthetic datasets respectively.

Table 2 and Table 3 obviously indicate that XProj algorithm is effective to distinguish the structural variations in documents, unfortunately, which may be limited if the XML documents exist significant differences on both structure and content. Similar to XProj, the performance of the SA algorithm is barely satisfactory in the situations where clustering only by structure features, such as ACM SIGMOD(2) and SYN1 datasets. The VSM algorithm that ignores the inherent structure information of XML documents is far inferior to other algorithms. Our proposed algorithm FXProj utilizing both structural and content features can be used to improve the performance of XML document clustering.

5.3 The Scalability of FXProj Algorithm

The next series of tests addresses FXProj's scalability with increasing numbers of documents and decreasing threshold of *min_sup*. When measuring FXProj's

Table 2. The Accuracy Comparison on the Real Datasets

DataSet	Method	F1
ACM SIGMOD(2)	FXProj	1
	XProj	1
	SA	0.97
ACM SIGMOD(5)	FXProj	0.91
	XProj	0.8
	VSM	0.47
	SA	0.62
INEX 2007	FXProj	0.81
	XProj	0.68
	VSM	0.29
	SA	0.51

Table 3. The Accuracy Comparison on the Synthetic Datasets

Data Set	Method	F1
SYN1	FXProj	1
	XProj	1
	VSM	0.75
	SA	0.98
SYN2	FXProj	0.93
	XProj	0.82
	VSM	0.6
	SA	0.75
SYN3	FXProj	0.89
	XProj	0.73
	VSM	0.49
	SA	0.69

scalability with increasing number of documents, we fixed the min_sup at 0.2. When measuring FXProj's scalability for decreasing threshold of min_sup , we fixed the number of XML documents. Figure 4 and Figure 5 indicate the scalability of FXProj Algorithm with increasing numbers of documents and decreasing threshold of min_sup respectively.

Figure 4 shows that the VSM algorithm and the other two algorithms are not of the same order of magnitude. Once the min_sup is lower than 0.3, VSM algorithm is incapable of clustering efficiently. Due to reducing a mass of redundant content information, FXProj algorithm requires little effort in Phase II, which ensures that FXProj and XProj are roughly equal in runtime.

Figure 5 demonstrates that there is a linear dependency of FXProj's processing time on the number of XML documents. This property means that the algorithm can easily deal with very large data sets.

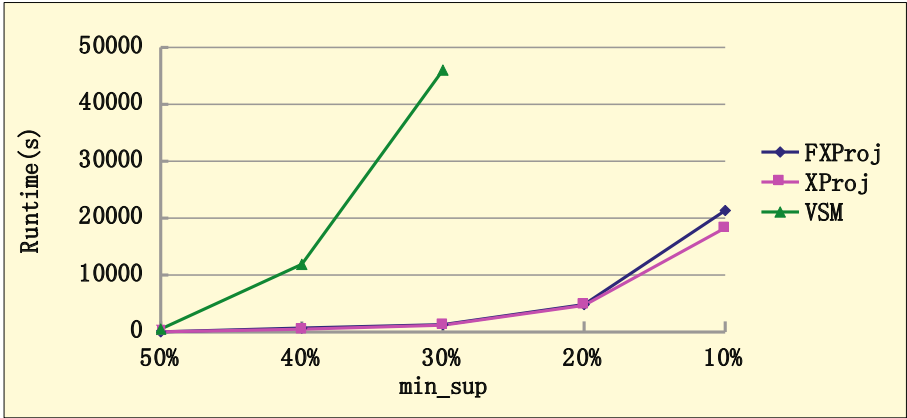


Fig. 4. Scalability Test of algorithms with decreasing threshold of min_sup

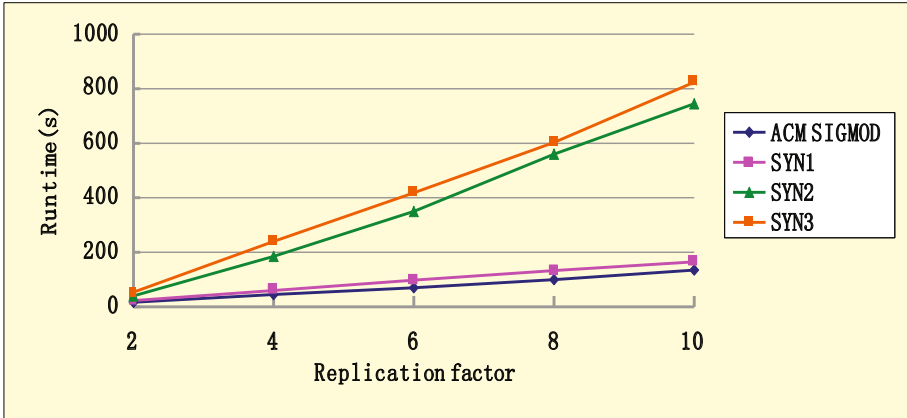


Fig. 5. Scalability Test of FXProj algorithm with increasing number of documents

6 Conclusion

The XProj algorithm is an efficient structure-only clustering algorithm, unfortunately, its accuracy are weakened because of ignoring the content information of XML documents. To address the shortcoming of XProj, we developed a new fuzzy clustering algorithm named FXProj. The main contributions are not only the combination of both structural features and content features, but also the adoption of some fuzzy techniques for XML documents clustering in a way that each frequent induced substructure has a fuzzy parameter associated with each cluster. The experimental results on both real datasets and synthetic datasets clearly ascertain that FXProj algorithm is capable of clustering XML documents

accurately and effectively. The scalability tests demonstrate FXProj algorithm is a scalable clustering method that can efficiently work for very large datasets.

Acknowledgment. This work was partially supported by National 863 program under Grant No. 2009AA01Z150. We would like to thank anonymous reviewers for their helpful comments.

References

1. Aggarwal, C.C., Ta, N., Wang, J., Feng, J., Zaki, M.: Xproj: a framework for projected structural clustering of xml documents. In: Proceeding of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 46–55 (2007)
2. Kutty, S., Nayak, R., Li, Y.: XCFS - An XML Documents Clustering Approach using both the Structure and the Content. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 1729–1732 (2009)
3. Seeland, M., Girschick, T., Buchwald, F., Kramer, S.: Online Structural Graph Clustering using Frequent Subgraph Mining. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS, vol. 6323, pp. 213–228. Springer, Heidelberg (2010)
4. Tran, T., Nayak, R.: Document Clustering using Incremental and Pairwise Approaches. *Focused Access to XML Documents*. 222–232 (2008)
5. Doucet, A., Ahonen-Myka, H.: Naive clustering of a large XML document collection. In: Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2002, pp. 81–87 (2002)
6. Kutty, S., Nayak, R., Li, Y.: XML Documents Clustering using Tensor Space Model A Preliminary Study. In: Proceedings of the 10th IEEE International Conference on Data Mining Workshops, ICDMW 2010, pp. 1167–1173 (2010)
7. Lesniewska, A.: Clustering XML Documents by Structure. In: Advances in Databases and Information Systems - Associated Workshops and Doctoral Consortium of the 13th East European Conference, ADBIS 2009, pp. 238–246 (2009)
8. Gan, G., Wu, J., Yang, Z.: The XML web: a first study. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003, pp. 500–510 (2003)
9. Hwang, J.H., Ryu, K.H.: A weighted common structure based clustering technique for XML documents. *Journal of Systems and Software*, 1267–1274 (2010)
10. Tekli, J., Chbeir, R., Yetongnon, K.: An overview on XML similarity: Background, current trends and future directions. *Computer Science Review*, 151–173 (2009)
11. Kutty, S., Nayak, R., Li, Y.: HCX: An Efficient Hybrid Clustering Approach for XML Documents. In: Proceedings of the 2009 ACM Symposium on Document Engineering, DocEng 2009, pp. 94–97 (2009)
12. Zhang, L., Li, Z., Chen, Q., Li, N.: Structure and Content Similarity for Clustering XML Documents. In: Shen, H.T., Pei, J., Özsu, M.T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., Shao, J. (eds.) WAIM 2010. LNCS, vol. 6185, pp. 116–124. Springer, Heidelberg (2010)
13. Domeniconi, C., Papadopoulos, D., Gunopulos, D., Ma, S.: Subspace clustering of high dimensional data. In: Proceedings of the SIAM International Conference on Data Mining (2004)

14. Abel, J., Teahan, W.: Universal Text Preprocessing for Data Compression. *IEEE Transactions on Computers*, 497–507 (2005)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 513–523 (1988)
16. Dalamagas, T., Cheng, T., Winkel, K.-J., Sellis, T.K.: Clustering XML Documents Using Structural Summaries. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) *EDBT 2004. LNCS*, vol. 3268, pp. 547–556. Springer, Heidelberg (2004)

Author Index

- Alhashmi, Saadat M. II-386
- Bain, Michael II-124
Bao, Xiaoyuan I-406
Benabdeslem, Khalid I-42
Berzal, Fernando II-166
Bisdorff, Raymond I-15
- Cai, Xiongcai II-124
Cai, Yanan I-97
Cao, Rongzeng II-359
Chawla, Sanjay II-237
Chen, EnHong I-175
Chen, Jinchuan II-96
Chen, Lisi I-311
Chen, Yashen I-367
Chen, Yueguo II-96
Compton, Paul II-124
Cubero, Juan-Carlos II-166
Cui, Bin I-381
- Danesh, Saeed I-162
Deng, Zhi-Hong I-395
Deris, Mustafa Mat I-299
Dobbie, Gillian I-285
Dong, Yuxiao I-97
Du, Liang I-215, II-372
Du, Xiaoyong II-96, II-266
Duan, Lei II-152
- Effantin, Brice I-42
Elghazel, Haytham I-42
Etoh, Minoru II-292, II-304
Eu-Gené, Siew II-386
- Feng, Ling I-138
Fournier-Viger, Philippe II-180
French, Tim I-162
Fu, Wanyu II-346
- Gao, Ning I-395
Gao, Peng II-359
Gao, Xiaoying II-55
Gollapalli, Mohammed II-252
Gong, Jibing II-69
- Gong, Shu II-110
Gong, Xueqing I-270
Gou, Chi II-152
Governatori, Guido II-252
Gu, Yanhui I-270
Guo, Shaosong II-82
- Hadzic, Fedja II-403
He, Jun II-266
Hecker, Michael II-403
Hu, Hao II-195
Huang, Congrui I-201
Huang, Shaobin II-346
Huang, Yongzhong II-138
Huang, Yuxin I-381
Hussain, Syed Fawad I-190
- Isa, Awang Mohd I-299
- Jabeen, Shahida II-55
Ji, Tengfei I-406
Jiang, Jia-Jian I-395
Jiang, Mengxia II-96
Jiang, Min II-152
Jiang, Shan I-201
Jiang, Xuan II-266
Jiménez, Aída II-166
Jin, Ping I-175
- Kang, Yong-Bin I-1
Kangkachit, Thanapat II-27
Katagiri, Masaji II-292
Ke, Qing I-97
Kechadi, M.-Tahar I-69
Kim, Yang Sok II-124
Koh, Yun Sing I-285
Kong, Shoubin I-138
Koper, Adam II-278
Krishnaswamy, Shonali I-1
Krzywicki, Alfred I-353, II-124
Kulczycki, Piotr I-152
Kuusik, Rein II-223
- Le-Khac, Nhien-An I-69
Li, Haifeng I-29

- Li, Jianhui II-1
 Li, Qiong I-229
 Li, Xuan I-215, II-372
 Li, Xue II-252
 Liang, Guohua I-339
 Limsetto, Nachai II-13
 Lind, Grete II-223
 Ling, Charles X. I-256
 Liu, Hongyan II-266
 Liu, Man I-124
 Liu, Mingjian I-243
 Liu, Wei I-162, II-237
 Lukasik, Szymon I-152
 Lv, Jing I-311
 Lv, Sheng-Long I-395
- Ma, Xiuli I-229
 Mahidadia, Ashesh II-124
 Mao, Yuqing I-109
 Meesuksabai, Wicha II-27
 Meyer, Patrick I-15
- Nguyen, Hung Son II-278
 Ni, Eileen A. I-256
 Nian, Jiazhen I-201
 Nie, Xinwei II-359
 Niu, Xiang II-138
- Olteanu, Alexandru-Liviu I-15
 Orchel, Marcin II-318
 Orimaye, Sylvester Olubolu II-386
 Ozaki, Tomonobu II-304
- Pan, Rong I-175
 Pang, Linsey Xiaolin II-237
 Pears, Russel I-285
- Qian, Weining I-270
 Qin, Zheng II-195
- Reynolds, Mark I-162
 Rose, Ahmad Nazari Mohd I-299
- Shen, Haifeng I-109
 Shen, Yi-Dong I-215, II-372
 Shi, Chuan I-97
 Shi, Shengfei I-311
 Song, Guojie II-359
 Sun, Chengzheng I-109
- Sun, Shengtao II-69
 Sun, Shiliang II-209
- Tagarelli, Andrea II-403
 Tan, Fei II-1
 Tang, Changjie I-325, II-152
 Tang, Shiwei I-229
 Tseng, Vincent S. II-180
- Waiyamai, Kitsana II-13, II-27
 Wang, Lidong I-55
 Wang, Shuliang I-367
 Wang, Tengjiao I-325
 Wang, Tingting II-266
 Wang, Wei II-332
 Wang, Yong I-243
 Wang, Yue I-325
 Wang, Zhichao I-83
 Wei, Baogang I-55
 Whelan, Michael I-69
 Wobcke, Wayne I-353, II-124
 Wood, Ian II-252
 Worawitphinyo, Phiradit II-55
 Wu, Bin I-97
 Wu, Jui-Yu II-41
 Wu, Liang II-1
- Xiao, Yu II-110
 Xie, Kunqing II-359
 Xie, Shuiyuan I-229
 Xiong, Xiaobing II-138
 Xu, Dafeng II-359
 Xu, Guandong I-175
 Xu, Ke II-138
 Xu, Wenhua II-195
 Xue, Anrong II-332
- Yan, Qiuling II-82
 Yang, Dongqing I-325, I-406, II-82
 Yang, Fenglei II-1
 Ye, Xia I-55
 Yin, Hongzhi I-381
 Yu, Hang I-395
 Yu, Hong I-124
 Yu, Jian II-110
 Yuan, Jie I-55
- Zaslavsky, Arkady I-1
 Zhang, Chengqi I-339
 Zhang, Huaxiang I-83

Zhang, Mingcai II-332
Zhang, Ning I-29
Zhang, Xing I-243
Zhang, Yan I-201
Zhang, Yanchun I-175
Zhang, Yang I-243
Zhang, Yin I-55
Zhang, Zhao I-270
Zhao, Bin I-270

Zhao, Nan II-195
Zhao, Ying II-346
Zheng, Yu II-237
Zhou, Aoying I-270
Zhou, Gang II-138
Zhou, Yuanchun II-1
Zhu, Jun I-325
Zong, Yu I-175
Zuo, Jie II-152